



République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université des Sciences et de la Technologie Houari Boumediène
Faculté d'Electronique et d'Informatique



Thèse

Présentée pour l'obtention du diplôme de Doctorat

EN : INFORMATIQUE

Spécialité : Intelligence Artificielle

par :

Saliha AOUAT

Sujet :

**Reconnaissance d'objets Tridimensionnels.
Approche Basée sur la Description Textuelle de
Silhouettes.**

Soutenue le 07/06/2009, devant le jury composé de :

Mme. H. DRIAS	Professeur, USTHB.	Présidente
Mr. S. LARABI	Professeur, USTHB.	Directeur de Thèse
Mme. L. HAMAMI	Professeur, ENP.	Examineur
Mr. A. DJERADI	Professeur, USTHB.	Examineur
Mr. M. DIAF	Professeur, UMMTO.	Examineur
Mr. S. AIT-AOUDIA	Maître de Conférence, ESI.	Examineur

A mes parents

Remerciements

Par l'intermédiaire de ces quelques lignes, je remercie Allah le tout puissant pour m'avoir donné le courage de faire ce travail.

Je tiens à adresser mes plus sincères remerciements à mon directeur de thèse, professeur Slimane LARABI pour son encadrement, sa disponibilité et pour ses conseils avisés.

Je tiens à remercier très vivement professeur Habiba DRIAS, Directrice du Laboratoire de Recherche en Intelligence Artificielle (LRIA), pour m'avoir fait l'honneur d'accepter de présider mon jury de thèse.

J'adresse également mes remerciements aux membres du jury : Mme Latifa HAMAMI Professeur à l'ENP, Mr Amar DJERADI Professeur à l'USTHB, Mr Moussa DIAF Professeur à l'UMMTO ainsi que Mr Samy AIT-AOUDIA Maître de conférence à l'ESI (ex INI) pour m'avoir fait l'honneur d'accepter de juger ce travail.

Je tiens aussi à remercier toutes les personnes que j'ai côtoyées durant toutes ces années, pour leur soutien dans les moments difficiles et l'ambiance amicale instaurée.

Je remercie tous mes collègues du laboratoire LRIA notamment les membres des équipes VISION et IACHM.

Mes pensées s'adressent bien sûr à ma famille, mes parents en particulier, sans qui cette thèse n'aurait pas vu le jour.

Résumé

Nous proposons dans cette thèse un ensemble de travaux réalisés dans le thème de la reconnaissance d'objets 3D à partir de leurs images. Le problème consiste à identifier à partir de l'image d'un objet requête les images des objets qui lui ressemblent à partir d'une base de descripteurs d'objets modèles. Les descripteurs des silhouettes modèles et requêtes sont textuels issus de leur décomposition en parties.

Notre contribution consiste à réaliser d'abord une réduction de l'espace de recherche de silhouettes similaires pour une requête donnée. Cette réduction, équivalente à un appariement grossier, se fait en deux niveaux : le premier consiste à indexer la base de descripteurs modèles, le second consiste à appliquer un ensemble de transformations sur la silhouette requête ou sur son descripteur afin de réduire les distorsions sur ses frontières. A noter que cette méthode peut être appliquée pour le suivi de formes dans une séquence d'images prises par une caméra en rotation.

Nous proposons ensuite une méthode pour affiner la mise en correspondance entre la silhouette requête et les silhouettes modèles. Deux approches sont proposées : pour la première approche, une mesure de similarité est définie par la comparaison des surfaces de leurs parties. Pour la seconde approche, les quasi-invariants sont utilisés pour la définition d'une mesure de similarité basée sur la comparaison des géométries des parties.

Les approches proposées sont applicables aux images du monde réel du fait de la prise en compte du bruit entachant les silhouettes extraites des images comme étant un élément principal. Cette assertion est prouvée par les résultats obtenus suite à leur application aux images du monde réel.

Mots clés : Reconnaissance de forme, Indexation, Mise en correspondance, Image requête, Image modèle, Base d'images modèles.

Table des matières

Résumé	vii
Introduction Générale.....	1
Chapitre 1	7
Etat de l'art sur la représentation et la reconnaissance d'objets 3D.....	7
1.1 Introduction	7
1.2 Généralités sur la vision artificielle.....	8
1.2.1 La vision humaine	8
1.2.2 La vision Artificielle	9
1.2.3 Description d'un système de vision artificielle	9
1.2.3.1 Systèmes de vision bidimensionnelle.....	10
1.2.3.2 Systèmes de vision tridimensionnelle	10
1.2.3.2.1 Systèmes actifs	10
1.2.3.2.2 Systèmes passifs	10
1.2.4 Le modèle de MARR	11
1.2.5 Traitements informatiques d'un système de vision.....	12
1.2.5.1 Traitements de bas niveau	12
1.2.5.1.1 L'acquisition d'images.....	12
1.2.5.1.2 Système d'acquisition d'images	12
1.2.5.1.3 Numérisation de l'image	13
1.2.5.1.4 Segmentation	13
1.2.5.2 Niveau intermédiaire	14
1.2.5.2.1 La stéréovision	14
1.2.5.3 Traitement de haut niveau	16
1.2.6 Le bruit dans une image	17
1.2.6.1 Filtrage de l'image	18
1.2.6.1.1 Les filtres linéaires	19

1.2.6.1.2 Les filtres non linéaires	19
1.3 Méthodes de représentation et reconnaissance d'objets.....	20
1.3.1 Origine et historique.....	20
1.3.2 Reconnaissance d'objets 3D : Complexité du problème.....	22
1.3.3 Reconnaissance d'objets 3D: Représentation et mise en correspondance	24
1.3.3.1 Représentations basées Objets	25
1.3.3.1.1 Représentations par courbure des frontières	25
1.3.3.1.2 Descriptions Axiales	27
1.3.3.1.3 Descriptions Surfaiques.....	28
1.3.3.1.4 Descriptions Volumiques	31
1.3.3.2 Représentations basées Apparence.....	33
1.3.3.2.1 Les graphes d'Aspect	34
1.3.3.2.2 Les Silhouettes	34
1.3.3.2.3 Analyse des composantes principales	36
1.3.3.2.4 Autres techniques	37
1.3.3.3 Récapitulatif de quelques représentations	38
1.3.3.4 Mise en correspondance (Matching).....	39
1.3.3.4.1 Hypothèse et vérification	39
1.3.3.4.2 Structures relationnelles	40
1.3.3.4.3 Regroupement des positions.....	41
1.3.3.4.4 Le hachage géométrique.....	42
1.3.3.4.5 L'arbre d'interprétation	43
1.3.3.4.6 Enregistrement rigide 3D	43
1.4 Conclusion.....	44
Chapitre 2	
Etat de l'art sur les techniques de représentation des silhouettes.....	47
2.1 Introduction	47
2.2 Extraction des silhouettes.....	48
2.2.1 Représentation d'une silhouette	48
2.2.2 Technique de traitement d'image.....	49
2.2.3 Technique de soustraction d'images	49
2.3 Classification des techniques de représentation des silhouettes.....	50
2.3.1 Les techniques de représentation de silhouettes basées sur les contours	51
2.3.1.1 Méthodes globales.....	52
2.3.1.1.1 Les descripteurs simples de silhouettes.....	52
2.3.1.1.2 Les signatures de silhouettes	53

2.3.1.1.3	Appariement élastique.....	53
2.3.1.1.4	La méthode stochastique	53
2.3.1.1.5	La méthode de l'espace d'échelle	54
2.3.1.1.6	La transformée spectrale	55
2.3.1.2	Méthodes structurelles.....	55
2.3.1.2.1	Représentation de chaîne de code	55
2.3.1.2.2	Décomposition en polygones	55
2.3.1.2.3	Analyse syntaxique	56
2.3.1.2.4	Les invariants de la silhouette	56
2.3.2	Les techniques de représentation de silhouettes basées sur les régions	57
2.3.2.1	Les méthodes globales	57
2.3.2.1.1	Moments géométriques invariants	57
2.3.2.1.2	Moments algébriques invariants.....	58
2.3.2.1.3	Moments descripteurs de silhouettes.....	58
2.3.2.1.4	Méthodes basées sur les descripteurs de grille.....	59
2.3.2.1.5	Matrice de la silhouette	59
2.3.2.2	Les méthodes structurelles	60
2.3.2.2.1	L'enveloppe convexe	60
2.3.2.2.2	L'axe médian.....	61
2.3.3	Utilisation de la silhouette pour la reconnaissance d'objets	62
2.4	Une approche textuelle pour la représentation des silhouettes	63
2.4.1	Description des Silhouettes suivant le langage LWDOS	64
2.4.1.1	Descripteur Global	65
2.4.1.2	Descripteur Détaillé.....	66
2.4.2	La grammaire du langage LWDOS.....	69
2.4.3	Le langage XLWDOS	73
2.5	Conclusion.....	77

Chapitre 3

Mise en correspondance des descripteurs textuels de silhouettes	79
3.1 Introduction	79
3.2 Effet de la déformation du contour d'une forme sur son descripteur LWDOS	80
3.3 Mise en correspondance des parties composées de descripteurs textuels	82
3.3.1 Objectif.....	82
3.3.2 Mise en correspondance des lignes de séparation principales	83
3.3.3 Mise en relation des parties composées	86
3.3.4 Réduction des parties composées associées aux lignes de séparation secondaires	89

3.4	Mise en correspondance des parties de descripteurs textuels	92
3.4.1	Principe de base.....	92
3.4.2	Règles de transformation.....	93
3.4.2.1	Cas d'un segment de droite additionnel	94
3.4.2.2	Cas d'une courbe additionnelle	96
3.5	Validation Expérimentale.....	98
3.5.1	Mise en correspondance des descripteurs globaux	99
3.5.2	Mise en correspondance des descripteurs détaillés	102
3.6	Conclusion.....	108
Chapitre 4		
	Reconnaissance des descripteurs textuels de silhouettes	109
4.1	Introduction	109
4.2	La description XLWDOS.....	110
4.2.1	Effet du bruit sur le descripteur XLWDOS.....	111
4.3	Reconnaissance globale de silhouettes.....	113
4.3.1	Principe de base.....	113
4.3.2	Mise en œuvre de la base de descripteurs de silhouettes modèles	113
4.3.2.1	Calcul des descripteurs de silhouettes	113
4.3.2.2	Indexation des descripteurs	114
4.3.3	Recherche d'une silhouette requête.....	116
4.3.3.1	La position du problème.....	116
4.3.3.2	Méthode de transformation du descripteur de la silhouette requête.....	117
4.3.3.3	Méthode de lissage de silhouettes	124
4.4	Mesure de similarité entre descripteurs requête et modèle	127
4.4.1	Mesure de similarité : Facteur global	128
4.4.1.1	Principe de base.....	128
4.4.2	Mesure de similarité : facteur local	133
4.4.2.1	Principe de base.....	133
4.4.2.2	Etude de variation des quasi invariants	134
4.4.2.2.1	Étude Statistique sur les éléments de l'index (θ, ρ)	138
4.4.2.2.2	Les seuils de comparaison pour la recherche	140
4.4.2.3	Application des quasi invariants sur les descripteurs détaillés XLWDOS	142
4.4	Etude de la complexité algorithmique.....	144
4.4.1	Rappel sur la complexité d'un algorithme	144
4.4.2	Complexité de l'algorithme de reconnaissance.....	146
4.5	Conclusion.....	147

Chapitre 5	
Validation de l'algorithme de reconnaissance	149
5.1 Introduction	149
5.2 Validation de l'algorithme d'indexation	150
5.2.1 Indexation d'une base d'images modèles.....	150
5.2.2 Recherche de forme dans la base de modèles par réduction de descripteurs	153
5.3 Mesures de similarité (calcul de surfaces)	155
5.4 Mesure de similarité (quasi invariants)	155
5.5 Autres expérimentations.....	163
5.5.1 Application sur la base d'images de Leibe et AL	163
5.5.2 Application sur la base de Mokhtarian.....	168
5.6 Conclusion.....	170
Conclusion Générale et Perspectives	173
Annexe A	
Lissage gaussien.....	177
A.1 Convolution avec le filtre gaussien	177
A.2 Extraction de contours.....	178
Annexe B	
Calcul des surfaces des parties XLWDOS.....	181
B.1 Calcul de la surface d'une partie	181
B.2 Exemple de calcul de surfaces.....	183
B.3 Exemple de classement de modèles.....	183
Bibliographie.....	185

Introduction Générale

Motivation

La vision a suscité l'intérêt de nombreux scientifiques et philosophes depuis très longtemps. Les neurophysiologistes mènent des recherches théoriques et expérimentales afin de comprendre l'anatomie et le fonctionnement du cerveau dans son ensemble ; ils ont découvert une structure très complexe qui est loin de leur révéler tous ses secrets.

Avec le développement des machines de calcul de plus en plus sophistiquées, des scientifiques se sont attaqués à la vision d'un point de vue quantitatif : est-il possible de construire un modèle calculatoire pour la perception visuelle? Il ne s'agit pas bien sûr de fournir une explication de comment fonctionne la vision biologique mais de créer un modèle, qui, vu de l'extérieur, possède des propriétés semblables.

La vision par ordinateur est une discipline dont les premières bases théoriques ont vu le jour dans les années 60. Depuis, étant donnée le spectre très large d'applications industrielles, militaires, aérospatiales et médicales qui peuvent être envisagées, la vision par ordinateur a vite dépassé le cadre relativement restreint des laboratoires de recherche.

La reconnaissance d'objets 3D est l'un des objectifs principaux de la vision artificielle. David Marr [139] propose dans son unique modèle de le résoudre en trois étapes :

- La prise de l'image de la scène observée et extraction des primitives pertinentes,
- Calcul des descriptions 2D ½ de la scène
- Calcul de la structure 3D des objets de la scène et leur reconnaissance.

La reconnaissance des objets moyennant une base d'objets modèles est l'approche classique dite *vision orientée modèle* (appelée aussi *approche basée objet*).

La seconde approche est la *vision par apparence* qui, comme son nom l'indique, s'appuie sur toutes les manifestations visuelles possibles des entités observées, c'est-à-dire, en théorie, sur toutes les images de ces entités prises de tous les points de vue et éclairages possibles [25]. Dans la première approche, les modèles se font dans le repère de la scène (ce sont donc des modèles 3D). En vision par apparence, les modèles sont construits dans le repère image (ce sont donc des modèles 2D).

La vision orientée modèle tente de résoudre un problème trop difficile pour être traité dans toute sa généralité. Si l'objectif est valide sur le plan scientifique, un modèle 3D du monde peut s'avérer inutile pour certaines applications. La modification du monde correspond à un changement d'état dans un espace de très grandes dimensions. Parmi ces dimensions, on trouve la nature des entités présentes dans le monde, leur position, leur orientation, leur forme (qui peut être variable dans le cas d'entités déformables), leur texture, la nature et la position des sources lumineuses, etc. Une recherche exhaustive des paramètres qui permettraient de reconstruire une image proche du stimulus source est vouée à l'échec en raison des coûts de calcul induits par le grand nombre de paramètres possibles. La vision par apparence constitue l'alternative.

La vision par apparence se fonde sur les manifestations visuelles d'entités d'intérêt captées par une caméra. *L'espace des apparences* d'une entité est l'ensemble des manifestations visuelles de cette entité prises selon tous les points de vue et conditions d'éclairage possibles. L'espace d'apparence est un concept abstrait et idéal. Il est impossible de décrire parfaitement toutes les apparences d'une entité, mais dans de nombreux cas, il est possible de se limiter aux manifestations d'apparences susceptibles d'être observées.

La classe des algorithmes fondée sur l'apparence des objets, réduit la tâche de la reconnaissance à un problème de mise en correspondance entre deux images, et ne se base sur aucune modélisation 3D. Ces algorithmes nécessitent la construction d'un système capable de classer les images par leurs contenus pour les stocker. Ceci sollicite les compétences des domaines du traitement d'images et des bases de données.

Le principe de la perception par apparence est de projeter l'espace d'apparences dans un espace de dimensionnalité réduite choisi de façon à faciliter les tâches à accomplir (indexation, reconnaissance, estimation de la position, etc.)

Parmi les méthodes basées sur l'apparence celles qui exploitent les silhouettes d'objets. En effet, dans le cas où l'objet est dépourvu des marques particulières de surfaces telles que les textures, on peut utiliser les silhouettes d'objets comme primitives d'images pour la représentation et la reconnaissance. Il s'agit en fait de décrire la forme d'une surface à travers le contour de celle-ci. Ces méthodes sont très intéressantes du moment qu'elles ne nécessitent pas beaucoup de place mémoire et sont assez rapides en temps d'exécution, ceci s'explique par la représentation d'un objet 3D uniquement par le contour de son aspect 2D d'où notre choix pour ce type de représentation notamment que les détails internes des objets ne seront pas pris en considération dans notre travail.

Contribution de la thèse

Les travaux effectués dans le cadre de cette thèse font partie de la seconde approche qui se propose de reconnaître les objets à partir de leurs apparences en utilisant les silhouettes.

Plusieurs algorithmes opérant sur les silhouettes pour la reconnaissance d'objets existent dans la littérature. Parmi ces méthodes, nous nous intéressons à une technique [120, 121] qui consiste à décrire la silhouette d'un objet avec du texte structuré reflétant la géométrie de sa frontière externe. Cette technique permet d'obtenir une bonne performance au niveau de l'espace mémoire nécessaire et au niveau du traitement.

Cependant, l'inconvénient de cette méthode est sa sensibilité au bruit et au changement de points de prise de vue. En effet même si deux silhouettes apparaissent identiques, il est fort possible que leurs descriptions soient très différentes. Une comparaison « brute » nous ramènerait à conclure sévèrement que les objets représentés sont différents. Comme les descripteurs obtenus à partir de cette description peuvent changer dans leurs structures à cause

de l'existence du bruit ou à cause d'un léger changement dans la prise de vue des images, ceci rend impossible leur utilisation dans la reconnaissance d'objets réels.

Dans le but de résoudre ce problème, nous proposons d'appliquer des transformations au descripteur requête ou bien un lissage de la frontière de la forme requête dans le but de trouver un sous ensemble de descripteurs modèles ayant le même index.

Les grandes lignes de notre approche sont décrites ci-dessous :

1) Le premier travail réalisé, est la mise en correspondance grossière des descripteurs [4, 5]. Pour ce faire nous nous sommes basés sur l'utilisation du langage LWDOS qui exprime les descripteurs comme un pourcentage relativement à l'unique rectangle minimum qui englobe l'objet. Pour effectuer cette comparaison nous développons des algorithmes permettant la transformation des descripteurs textuels en prenant en considération les sous descripteurs dus au bruit [6, 7, 9, 12].

2) La transformation et la comparaison des descripteurs textuels étant possible, nous proposons ensuite une méthode de reconnaissance d'une image requête en disposant d'un ensemble de descripteurs d'images modèles. Pour mener à bien la tâche de la reconnaissance, plusieurs points doivent être résolus :

- Mise en œuvre de la base de descripteurs modèles et leur indexation [8, 13].
- Deux méthodes sont proposées pour la recherche de l'index du descripteur requête dans la table des index des descripteurs modèles en tenant compte de la déformation de la silhouette requête. La première basée sur la réduction de son descripteur et la seconde basée sur le lissage de la frontière de la silhouette correspondante [119, 118, 13].
- Enfin nous proposons des mesures de similarité définies à la base des surfaces des parties [10], ainsi que les quasi-invariants géométriques pour la reconnaissance [11]. Nous exploitons dans cette étape, les descripteurs textuels détaillés.

Organisation de la thèse

Cette thèse est organisée en deux grandes parties : **l'état de l'art** dans les chapitres 1 et 2 et **la contribution de la thèse** ainsi que la **validation des algorithmes** dans les chapitres 3, 4 et 5.

- Dans le **premier chapitre** nous commençons par les notions élémentaires sur la vision artificielle et les étapes suivies par le processus de vision à partir de l'acquisition de l'image jusqu'à l'interprétation de celle-ci. Nous présenterons ensuite un état de l'art sur les méthodes de la représentation et de la reconnaissance d'objets 3D. Il s'agira de deux grandes techniques existantes pour la représentation et la reconnaissance ; ce sont les techniques basées objets et celles basées sur l'apparence des objets.
- Le **deuxième chapitre** présentera un état de l'art sur les méthodes existantes dans la littérature et qui utilisent les silhouettes pour représenter et reconnaître les objets. Nous résumerons, en particulier, la description textuelle sur laquelle se base nos algorithmes ; il s'agira de deux descriptions LWDOS et XLWDOS permettant la représentation des silhouettes d'objets avec un code textuel [120, 121].
- Nous présenterons dans le **troisième chapitre** la première contribution de la thèse concernant l'appariement des descripteurs globaux et détaillés selon le langage LWDOS [120] et ceci en développant des algorithmes de réduction et d'approximation de descripteurs pour leur mise en correspondance. La validation expérimentale de cette méthode sera présentée aussi dans ce chapitre.
- Le **chapitre quatre** est consacré à notre méthode de reconnaissance basée sur le langage XLWDOS [121]. Nous commencerons par la mise en œuvre de la base de descripteurs modèles et leur indexation. Deux méthodes sont proposées pour la recherche de l'index du descripteur requête dans la table des index des descripteurs modèles en tenant compte de la déformation de la silhouette requête. La première

basée sur la réduction de son descripteur et la seconde basée sur le lissage de la frontière de la silhouette correspondante. La dernière étape consistera à définir des mesures de similarité des descripteurs globaux et détaillés afin d'affiner et de compléter le processus de la reconnaissance.

- Le **cinquième chapitre** concernera la validation expérimentale de l'algorithme de reconnaissance, la validation de chaque étape sera présentée commençant par l'indexation jusqu'à la reconnaissance complète basée sur les mesures de similarité.

Une conclusion générale ainsi qu'un ensemble de perspectives aux différents travaux menés, termineront cette thèse.

Chapitre 1

Etat de l'art sur la représentation et la reconnaissance d'objets 3D

1.1 Introduction

La majorité de ce qui nous entoure dans l'espace a une structure tridimensionnelle et sa description se fait par l'homme sans aucune difficulté, pourtant l'information qui est réellement présente sur la rétine n'est rien d'autre qu'une collection de points. En chaque point ou pixel se trouve une information qui donne une indication sur la quantité de lumière et des couleurs provenant de l'environnement extérieur et qui ont été projetées sur la rétine. Le souci majeur des chercheurs en vision artificielle est d'essayer de reproduire ces fonctions sur des machines afin d'interpréter automatiquement des images digitalisées.

L'interprétation est un processus qui fait partie du système de vision. Aussi, il n'y a pas une correspondance de manière précise entre l'information sensorielle (lumière et couleur) et le résultat que fournit ce processus. Ceci étant dit, des connaissances nécessaires doivent être fournies afin de permettre une interprétation sans ambiguïté [95]. Aujourd'hui, l'être humain tente de reproduire la vision humaine par un système similaire. Ceci a donné naissance à ce qu'on appelle « **vision par ordinateur** » ou « **vision artificielle** » [200], celle ci représente une discipline de l'intelligence artificielle.

La reconnaissance d'objets attire l'attention de plusieurs chercheurs en vision artificielle, c'est une étape nécessaire pour le développement d'un système de vision complet utilisable dans plusieurs domaines d'applications et capable de prendre des décisions pour l'accomplissement des tâches spécifiques [184,59].

Nous présentons dans ce chapitre des généralités sur la vision artificielle suivi d'un état de l'art sur les différentes méthodes de représentation et de reconnaissance d'objets 3D.

1.2 Généralités sur la vision artificielle

La vision par ordinateur est devenue un vaste domaine de recherche qui tente de remplacer la vision humaine par des machines plus ou moins indépendantes. Ces dernières peuvent non seulement prendre des images, mais surtout assurer leur compréhension. Autrement dit, assurer la même interprétation que fournit le système visuel humain et surtout arriver aux mêmes résultats. Avant d'étudier la vision par ordinateur, nous commencerons par expliquer brièvement le processus de la vision humaine.

1.2.1 La vision humaine

La vision humaine est un système complexe et sophistiqué. Elle implique des processus physiologiques, biochimiques, neurologiques et psychologiques.

La vision est un sens très développé chez l'homme. Les informations visuelles traitées par le cerveau ont un rôle déterminant pour la perception de l'environnement, les déplacements, et la vie professionnelle ou relationnelle.

L'œil de l'être humain reçoit une quantité très importante d'informations. Il est très sensible aux spectres des ondes électromagnétiques (lumière visible). Aussi, tout comme chaque être humain est différent de son semblable, la vision l'est aussi. Mais dans tous les cas, elle autorise une vision stéréoscopique (c'est-à-dire 3D). La vitesse de déplacement, la distance, la couleur et l'évaluation, sont des informations transmises au cerveau par le nerf optique. Ces rayons lumineux se caractérisent par environ un million de points reçus par la rétine [40]. Ces points contiennent des informations sur la quantité de lumière et de couleurs provenant de l'environnement. Ils forment l'image qui sera conduite sous forme de signaux

transmissibles le long des voies neuronales vers l'appareil cortical d'analyse. A ce stade, le cerveau prend le relais et donnera une signification à l'image perçue par l'œil.

La performance visuelle est donc un élément fondamental de la qualité de vie. La performance visuelle peut être appréciée de deux façons :

- quantitativement (acuité visuelle),
- qualitativement (perception des contrastes, vision nocturne, résistance à l'éblouissement, sens du relief, des vitesses de déplacement, perception des couleurs).

1.2.2 La vision Artificielle

La vision par ordinateur est constituée de l'ensemble des technologies de traitement de l'information numérique ayant pour origine des capteurs photométriques [75, 43, 69, 17].

Un système de vision artificielle prend en entrée une ou plusieurs images numériques et effectue des traitements sur ces images afin d'en extraire des informations relatives à la scène observée.

Contrairement à la perception humaine qui est basée sur deux images (œil droit et œil gauche), l'ordinateur peut quant à lui utiliser des sources d'informations plus nombreuses et plus variées, éventuellement dans d'autres bandes spectrales que le visible [1].

1.2.3 Description d'un système de vision artificielle

Un système de vision artificielle est un ensemble de traitements informatiques ayant pour tâche la reproduction de la fonction « voir ». Ce système reçoit en entrée une image comme information et en sortie une description symbolique de cette image, entre les deux se situe tout un ensemble de traitements qui doivent être effectués afin de réaliser la transformation des informations demandées. On trouve dans les systèmes de vision deux catégories distinctes :

- Systèmes bidimensionnels.
- Systèmes tridimensionnels.

1.2.3.1 Systèmes de vision bidimensionnelle

Ce sont les systèmes qui n'incluent pas la notion de profondeur ; c'est à dire ils ne peuvent reconnaître que des objets plans puisqu'ils n'utilisent que des données à deux dimensions. Ces systèmes trouvent place dans la télédétection, la biomédicale, la reconnaissance des caractères, etc.

1.2.3.2 Systèmes de vision tridimensionnelle

Ces systèmes sont pour leur part embarqués sur les robots mobiles évoluant en scènes d'intérieur ou d'extérieur, les dotant ainsi d'un sens qui contribue à une autonomie dans la prise de décision imitant pour cela le système visuel humain comme par exemple : la saisie d'objets, l'évitement d'obstacles.

Les systèmes de vision sont classés en deux catégories :

- Systèmes actifs.
- Systèmes passifs.

1.2.3.2.1 Systèmes actifs

Ils prennent en charge le contrôle de l'éclairage de la scène. Pour cela des capteurs de type laser (télémètre laser) ou ultrason sont utilisés. Ces capteurs disposent d'un émetteur et d'un récepteur qui acquière une information spatiale. En ce qui concerne leur analyse, il en existe deux types :

- **La triangulation** : le principe se fonde sur la connaissance de la distance entre le récepteur et l'émetteur, ainsi on peut déduire la profondeur du point qui réfracte l'onde de la direction du signal émis et réfracté.
- **Le temps de vol** : on peut calculer la profondeur du point par la mesure du temps mis par le signal entre l'émetteur et le récepteur.

1.2.3.2.2 Systèmes passifs

Contrairement aux systèmes de vision actifs, les systèmes passifs ne contrôlent pas l'éclairage de la scène. Les seules données disponibles sont les images, donc le degré de

complexité est élevé par rapport aux systèmes actifs. Ils traitent une ou plusieurs images 2D prises par des caméras et contenant une information de luminance (niveau de gris ou couleur) émise ou réfléchi par l'environnement.

1.2.4 Le modèle de MARR

La philosophie de David Marr est détaillée dans la présentation de *Vision* [139]. Comme l'a signalé Gordon [78], la contribution de D. Marr au domaine de la perception visuelle est la résultante des recherches en intelligence artificielle, théorie de l'information et informatique. D'après D. Marr, la notion *d'information* était plus large que le seul sens quantitatif que lui voit la théorie de l'information. De la même manière, il entendait par processus de traitement toute transformation d'une représentation en une autre représentation.

Vers la fin des années 70, David Marr [139] a proposé un modèle computationnel pour le traitement et la représentation de l'information visuelle. Son modèle consiste à reconstruire une scène tridimensionnelle à partir des données bidimensionnelles. Il a ainsi réalisé la décomposition du processus de la vision artificielle en processus où chacun d'eux réalise une étape particulière.

Le modèle qui a été proposé est un modèle calculatoire pour le traitement et la représentation des informations visuelles, les principaux traits du paradigme de D. Marr sont [139]:

- Le premier est un processus d'extraction de caractéristiques qui produit une description en termes d'attributs bidimensionnels à partir d'une image. Ce niveau de représentation est connu sous le nom : première ébauche.
- Le second est un ensemble de processus plus ou moins indépendants afin de déterminer les propriétés tridimensionnelles locales relatives à la scène. En fait c'est une représentation centrée sur l'observateur appelée **ébauche 2.5D**. Ce processus peut opérer sur une seule image, une paire d'images (stéréoscopie) ou une séquence d'images (analyse du mouvement). Le processus opérant sur une seule image est un processus d'inférence en utilisant des connaissances géométriques (analyse des contours),

géométriques et statistiques (analyse des textures), photométriques (analyse des ombres) ou colorimétrique (analyses des reflets).

- Le dernier met en correspondance l'ébauche 2.5D avec les connaissances 3D pour construire une description de la scène en termes d'objets et de relation entre eux. Il s'agit d'une représentation centrée sur la scène.

1.2.5 Traitements informatiques d'un système de vision

1.2.5.1 Traitements de bas niveau

Ils consistent principalement à acquérir les données de base et les préparer pour les niveaux supérieurs, on peut les partager en deux étapes principales : l'acquisition de l'image et sa segmentation.

1.2.5.1.1 L'acquisition d'images

L'acquisition d'images est la transformation du signal lumineux vers une image en niveaux de gris ou couleur. Une image de niveaux de gris est codée sous forme de tableau à deux dimensions, chaque élément du tableau appelé : pixel (Picture element) est un entier codant l'intensité lumineuse perçue en un point.

1.2.5.1.2 Système d'acquisition d'images

D'après CASTAN [43], un système d'acquisition d'images est composé des éléments suivants :

- Un objet à examiner.
- Un dispositif optique.
- Un système transformant l'énergie lumineuse en énergie électrique.
- Un système d'éclairage (contrôlé ou pas).
- Un système permettant d'échantillonner le signal vidéo.
- Une mise en forme et un stockage de l'image numérisée.

La lumière est réfléchiée par les objets présents dans la scène à travers la lentille de la caméra. Les photons porteurs de luminosité sont décelés par les détecteurs de la caméra, il y a alors la génération de charges électriques. Les charges obtenues sont interprétées en termes de

voltages qui sont échantillonnées [69]. L'image qui se forme sur la zone photosensible de la caméra est représentée par une fonction bidimensionnelle $F(x,y)$ où x et y sont les coordonnées spatiales. La valeur de cette fonction en un point est proportionnelle à l'intensité lumineuse en ce même point, il s'agit de la fonction de niveaux de gris.

1.2.5.1.3 Numérisation de l'image

L'image analogique obtenue par le système d'acquisition ne peut pas être stockée directement en mémoire car le signal analogique est continu. De ce fait, on tire les données digitales qui correspondent à la discrétisation des intensités lumineuses et des coordonnées spatiales des points de l'image. Pour les images monochromes l'intensité lumineuse de chaque point est représentée sur huit bits, par contre pour les images en couleur elle est représentée sur $3*8$ bits (24bits).

Une fois cette image digitalisée, elle peut être présente sous forme d'une image en niveaux de gris où le stockage d'un point se fait sur huit bits ce qui nous donne 256 valeurs pour la codification des intensités lumineuses des pixels de l'image, où la valeur 0 représente le noir tandis que 255 représente le blanc, et les valeurs intermédiaires correspondent aux différentes teintes du gris.

1.2.5.1.4 Segmentation

A l'issue de la segmentation, l'image est représentée par un ensemble de primitives. Ces dernières peuvent être des pixels, des segments de contours ou des régions.

a) Segmentation par contour

Elle nécessite la détection de contours contenus dans l'image, la détection de contours se ramène à la recherche des discontinuités locales de la fonction des niveaux de gris de l'image.

La segmentation par contours consiste en une approximation des contours par des représentations analytiques telle que des droites.

La difficulté majeure pour la détection de contour réside en la présence du bruit dans l'image qui peut provoquer la détection de faux points de contour.

b) Segmentation par région

Elle est duale à la segmentation par contour. En effet, au lieu de rechercher les points de discontinuité, on recherche des continuités de certaines propriétés sur des groupes de pixels adjacents, les pixels ne sont pas traités individuellement mais en groupes de pixels contigus liés par une propriété.

1.2.5.2 Niveau intermédiaire

A ce stade de traitement, avec l'utilisation des données du niveau précédent, on peut faire une reconstruction tridimensionnelle nécessitant l'appariement et la triangulation des images stéréoscopiques.

1.2.5.2.1 La stéréovision

On définit la vision stéréoscopique comme étant un processus qui permet de retrouver une image en relief (structure tridimensionnelle) à partir de deux images [192]. Elle est dite statique lorsque le positionnement relatif des objets est inchangé sur les points de vues, et dynamique lorsqu'il y'a mouvement.

MARR et POGGIO [139] proposent une approche pour la résolution du problème de la vision, qui consiste à suivre la démarche suivante :

- Localisation des primitives pertinentes.
- Calibrage du système de prise de vue.
- Mise en correspondance des primitives (Appariement).
- Détermination de la structure 3D (Triangulation).

Nous expliquons ci-dessous chacune de ces étapes:

a) Choix des primitives (extraction des primitives pertinentes)

Le choix des caractéristiques visuelles (primitives) se fait à l'issue de la segmentation des deux images stéréoscopiques. Parmi ces primitives, citons la plus simple qui est le pixel, ensuite les segments de contours et les régions.

b) Calibration de caméras

Cette étape est nécessaire quels que soient les capteurs utilisés, et plus encore dès l'instant où leurs informations doivent être mises en commun. Elle permet d'exprimer les informations issues de chaque capteur par rapport à un référentiel commun en vue de localiser dans un environnement des objets ou des obstacles.

c) Appariement (mise en correspondance)

L'appariement est une opération qui consiste à établir une correspondance entre les deux ensembles de primitives issues des deux images à la suite de la segmentation (appairier point à point les deux images). Cette action est une tâche très délicate en stéréovision, pour cette raison plusieurs contraintes ont été mises au point pour réduire la complexité et l'ambiguïté, telles que les contraintes géométriques (la contrainte épipolaire par exemple), et qui ont été renforcées pour leurs insuffisances par les contraintes suivantes :

*** La contrainte de disparité**

La contrainte de disparité est issue du calibrage du capteur. Lorsqu'on dispose d'un ensemble statique caméra-scène, on peut évaluer à partir d'images tests les disparités minimum et maximum qui apparaîtront dans une paire d'images de cette scène, on pourra ainsi limiter l'espace de recherche des points homologues sur les deux images.

*** La contrainte épipolaire**

La contrainte épipolaire réduit l'espace de recherche des correspondants d'un espace 2D (image entière) à un espace 1D (droite épipolaire). Les recherches ont défini l'épipole gauche (resp. droit), comme étant l'intersection de la droite passant par les centres focaux et le point image gauche (resp. droit) avec le plan image droit (resp. gauche). Les deux projections obtenues du point choisi sont situées sur les deux droites épipolaires.

*** La contrainte d'unicité**

On formule par cette contrainte l'unicité du point homologue, ce qui revient à dire que chaque point possède au plus un correspondant sur l'autre image. Cette contrainte est respectée si les objets de la scène sont opaques.

*** La contrainte d'ordre**

Cette contrainte présume que l'ordre de projection des points sur les deux images est conservé. De ce fait si un point M1 est à droite d'un point M2 sur l'image gauche, il sera de même pour leurs correspondants sur l'image droite.

d) Triangulation

La triangulation consiste à reconstruire les coordonnées d'un point donné dans l'espace 3D à partir de ses projections sur les images calculées suite à l'opération de l'appariement.

1.2.5.3 Traitement de haut niveau

Les traitements de haut niveau, ont pour objectif l'interprétation des données du niveau intermédiaire, afin de permettre la reconnaissance de la scène observée.

Reconnaître un objet signifie lui associer un modèle appartenant à une base de modèles. Une tendance actuelle est d'envisager des systèmes de vision dits « intelligents » essayant non pas de reconnaître un objet élémentaire en tant que tel, mais plutôt d'effectuer l'analyse d'une scène complexe. On définit alors une scène comme un ensemble d'objets entre lesquels existe un ensemble de relations à des niveaux sémantiques divers. On distingue deux grandes classes de traitement de haut niveau [141] :

La première classe est propre à tous les systèmes dépourvus de la description des surfaces des objets (absence du niveau intermédiaire). Elle consiste à comparer la description symbolique d'une sous image avec les structures issues des images des modèles, pris sous des angles de vue différents. Lorsqu'une ressemblance suffisante a été détectée, le modèle correspondant est sélectionné, et si l'image du modèle choisi est voisine de l'image initiale, on estime alors que la reconnaissance est bonne.

La deuxième classe de traitements utilise la description des surfaces fournies par le traitement intermédiaire pour les décomposer en primitives surfaciques. Ces primitives seront représentées par des primitives volumétriques (à titre d'exemple les cylindres généralisés), et on utilise les relations spatiales trouvées pour sélectionner un modèle.

1.2.6 Le bruit dans une image

Le bruit représente les parasites ou les interférences d'un signal, c'est-à-dire les parties du signal déformées localement. Ainsi le bruit dans une image désigne les pixels de l'image dont l'intensité est très différente de celles des pixels voisins, ou une brusque variation des niveaux de gris des pixels [17, 95]. Le bruit peut être très désagréable à la vision ou à l'analyse d'une image, il occasionne des résultats inattendus abaissant ainsi les performances du système car ceci a pour effet l'altération de l'apparence et la signification même de l'image, il faut donc l'atténuer voire le supprimer.

Pour l'acquisition d'une image, plusieurs instruments sont mis en œuvre, le plus souvent sont des caméras. Des effets internes liés au matériel d'acquisition ou externes peuvent causer la déformation de l'image prise, c'est-à-dire changer l'apparence et parfois même des fois la signification de l'image à cause de certains pixels qui s'ajoutent ou se soustraient en modifiant l'intensité lumineuse. En terme de fréquence, le bruit est une modulation à très haute fréquence de l'image. Il peut être causé par :

- Une instabilité de la source du signal.
- Un signal parasite provenant d'ailleurs.
- Une instabilité du récepteur.

Lorsque le bruit s'intrus sur une image, il la déforme et fausse les résultats de son analyse de même pour la reconnaissance de la forme d'un objet. Ainsi les performances d'un système de vision se dégradent. La seule solution est de trouver des moyens pour le supprimer et atténuer son effet désagréable.

1.2.6.1 Filtrage de l'image

Pour améliorer la qualité visuelle d'une image, on doit éliminer les effets du bruit (parasites) en lui faisant subir un traitement appelé filtrage [17].

Le filtrage consiste à appliquer une transformation (appelée filtre) à tout ou une partie d'une image numérique en appliquant un opérateur. On distingue à titre d'exemple les filtres suivants :

- Les filtres passe-bas : consistant à atténuer les composantes de l'image ayant une fréquence haute (pixels foncés). Ce type de filtrage est généralement utilisé pour atténuer le bruit de l'image, c'est la raison pour laquelle on parle habituellement de lissage.
- Les filtres passe-haut, à l'inverse des filtres passe-bas, ils atténuent les composantes de basse fréquence de l'image et permettent notamment d'accentuer les détails et le contraste, c'est la raison pour laquelle le terme de "filtre d'accentuation" est parfois utilisé.
- Les filtres passe-bande permettant d'obtenir la différence entre l'image originale et celle obtenue par application d'un filtre passe-bas.

Une convolution a pour entrée les niveaux de gris des pixels recouverts par la fenêtre d'une part et les valeurs des coefficients d'autre part et pour résultat une nouvelle valeur de niveau de gris. Ce nouveau paramètre est déterminé par le déplacement de la fenêtre et le remplacement de la valeur de chaque pixel par le résultat de l'opération appliquée à ses voisins. Ainsi une nouvelle image est produite.

On remarque qu'en pratique, la convolution numérique d'une image se fera simplement par une sommation de multiplications.

On peut catégoriser les filtres en deux ensembles : les filtres linéaires et les filtres non linéaires.

1.2.6.1.1 Les filtres linéaires

On appelle ce type de filtres « linéaires » parce qu'ils utilisent une combinaison linéaire des valeurs du voisinage pour déterminer le niveau de gris d'un pixel.

Les filtres linéaires les plus utilisés sont : le filtre de moyenne et le filtre gaussien.

* Le filtre de moyenne :

Le filtre de moyenne est une version d'une gamme plus vaste de filtres de lissage dits linéaires qui remplacent le pixel courant par la valeur moyenne des pixels sur une fenêtre.

* Le filtre de Gauss :

Le filtre gaussien est un filtre passe-bas. Il est utilisé pour améliorer la qualité de l'image. Il a pour effet d'adoucir l'image. Ce filtre utilise des matrices de convolution de taille impaire (3x3 ou 5x5 ou plus). Une convolution est un traitement d'une matrice mathématique par une autre appelée matrice de convolution ou «noyau». Elle est caractérisée par un masque qui est une matrice de coefficients.

Ce filtre est une opération de convolution d'une image avec une gaussienne :

$$G(x,y,\sigma) = (1/\sigma\sqrt{2\pi})e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

Le filtre étudie successivement chacun des pixels de l'image. Pour chaque pixel, il multiplie la valeur en niveau de gris de ce pixel avec celles des 8 pixels qui l'entourent par la valeur correspondante dans le noyau. Il additionne l'ensemble des résultats et le pixel initial prend alors la valeur du résultat final.

1.2.6.1.2 Les filtres non linéaires

* Le filtre médian :

Cette opération consiste à classer les pixels voisins par ordre croissant des valeurs de niveaux de gris, puis affecter la valeur médiane au pixel central.

Ce filtre paraît très efficace pour une image dégradée par une source de bruit impulsionnel. Donc pour les variations brusques, il permet d'éliminer les pixels isolés.

*** Les filtres morphologiques :**

Ce type de filtre est basé sur deux opérations :

- **L'érosion** : On parcourt une fenêtre de 3x3 pixels, on fait un « ET logique » des huit voisins du pixel central. Cette opération permet de diminuer le contour de l'objet avec l'épaisseur d'un pixel.
- **La dilatation** : C'est l'opération duale de l'érosion. Elle effectue un « OU logique » des huit voisins du pixel central.

1.3 Méthodes de représentation et reconnaissance d'objets

Les chercheurs en vision artificielle ont développé deux principales approches pour résoudre le problème de la représentation et la reconnaissance d'objets :

La première approche est dite ascendante (Bottom-up), elle suit dans son principe la théorie de Marr [139] aboutissant à une reconstruction puis à l'interprétation de scènes à partir d'une ou plusieurs images, en passant par l'appariement, la triangulation et le calibrage des caméras.

La deuxième approche est descendante (Top-down), elle essaye d'éviter la complexité de reconstruction de scènes, elle se base en fait sur le principe que la reconnaissance n'est pas conditionnée par une obtention préalable de la carte de profondeur. Cette approche compte sur la présence d'objets ou de familles d'objets dans une base de modèles puis développe des algorithmes de mise en correspondance (matching) entre l'objet requête (un objet de la scène) et les objets modèles.

1.3.1 Origine et historique

Les fondements de la reconnaissance en vision par ordinateur remontent au début des années 1960, lorsque les premiers travaux ont été effectués dans des domaines très simples.

Le monde était modélisé comme étant composé d'un ensemble de blocs polyédriques définis par les coordonnées de leurs sommets et de leurs bordures [19]. Chaque bloc représentait une zone homogène de l'image ayant la même luminosité, et chaque bordure n'est autre que les zones de discontinuité de l'intensité lumineuse. Cette notion a été reprise et maintenue par les applications ultérieures de la vision. Larry Roberts [173] a été le premier à développer un système dont la première étape consistait à segmenter l'image en un ensemble de segments de droites (produisant un dessin de lignes). Cette information a ensuite été utilisée pour l'élaboration des prototypes 3D.

Cependant plusieurs tâches effectuées facilement par le système visuel humain, telles que la perception de l'ombre ou encore la réflexion de l'intensité par les surfaces des objets posaient d'énormes problèmes dans ces premiers systèmes de vision.

Durant les années 1960 à 1970 les systèmes développés intégraient la segmentation et aussi l'interprétation. Des exemples classiques tels que [66] reconnaissent les objets à la suite d'un appariement partiel avec les modèles, puis la méthode de vérification d'hypothèse est employée pour effectuer une interprétation globale sur l'image candidate.

En 1978, David MARR [139, 140] proposait une approche rigoureuse connue par la suite sous le nom du paradigme de Marr. Cette approche consiste à décomposer le processus de vision en trois étapes essentielles : le bas niveau (extraction des primitives), le niveau intermédiaire (l'analyse des surfaces, la stéréovision et la reconstruction 3D) et le haut niveau (interprétation).

Les années 1980 ont vu l'apparition des systèmes de vision actifs, ces systèmes nécessitent le contrôle de l'éclairage de la scène et les conditions des prises de vues [185, 2, 32].

Pendant ces dernières années, beaucoup de travaux ont été entrepris pour développer la vision artificielle [18]. Nous proposons dans ce chapitre de présenter les principales méthodes.

1.3.2 Reconnaissance d'objets 3D : Complexité du problème

Un système de reconnaissance d'objets débute par l'acquisition d'images 2D à partir de scènes 3D via des transformations orthogonales ou perspectives. Après ce processus de traitement de bas niveau qui consiste en plus à corriger la qualité de l'image par un ensemble d'opérations de lissage et de réduction du bruit, l'image est traitée afin de refléter la position des objets dans la scène ainsi que la relation entre ces objets [59].

La modélisation est constituée de deux phases :

La première est hors ligne (off line) qui consiste à représenter les images de la base qui puissent exister dans la scène et construire ainsi une base de modèles.

La deuxième est une étape en ligne (on line) qui est l'étape de la reconnaissance proprement dite elle consiste à effectuer des transformations spatiales, à déterminer la position d'un objet dans la scène et surtout à réaliser une mise en correspondance entre les objets de la scène et les modèles de la base d'images construites lors de l'étape hors ligne [3, 52].

Cette reconnaissance très populaire en vision est appelée reconnaissance basée modèles (model-based object recognition).

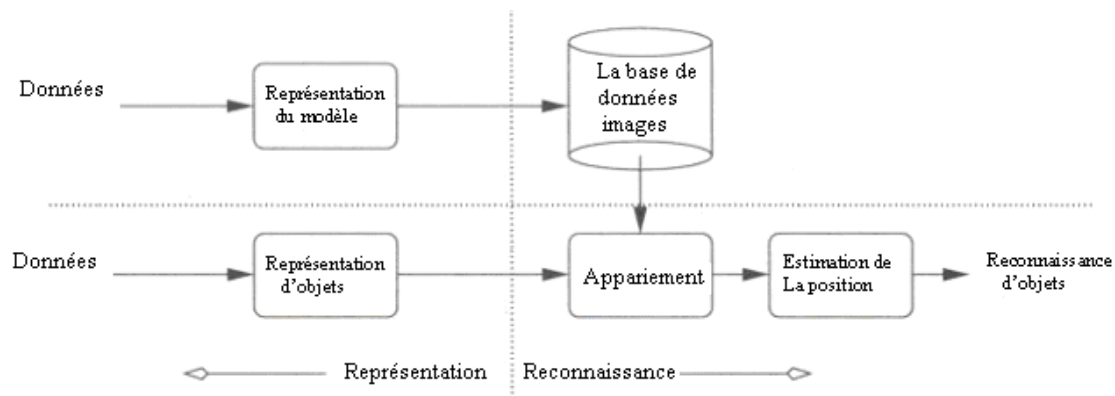


Figure 1.1 La reconnaissance d'objets basée modèle

Plusieurs chercheurs pensent que la détermination de la position et de la carte de profondeur est une condition préalable pour la reconnaissance. Cependant d'autres chercheurs notamment dans la tendance actuelle en vision artificielle s'opposent à cette contrainte.

David LOWE défend la possibilité de reconnaître un objet sans être contraint à effectuer une reconstruction préalable de l'objet [132], il se base sur le fait que l'homme est capable de reconnaître un objet et de percevoir toute la scène avec un seul œil. L'utilisation d'un système 3D se révèle utile uniquement lorsque l'estimation de la position et la détection d'obstacles est nécessaire, par contre lorsqu'il s'agit seulement d'analyser une image et dire quels objets elle représente, un système 2D suffit, ce dernier consistera à comparer une image 2D (image requête) avec tous les aspects 2D d'une image modèle stockés dans la mémoire du système, ce qui est couramment appelé dans la littérature « reconnaissance par apparence ».

La reconnaissance automatique d'objets est un problème difficile à résoudre pour plusieurs raisons entre autre :

- 1) Les détails du système visuel humain ne sont pas complètement interprétés ; les neurophysiologistes ignorent beaucoup de détails sur les processus suivis par le cerveau afin d'interpréter les données observées. Pour cela il est difficile de fournir une description mathématique de ces processus, qui sont souvent décrits superficiellement par les algorithmes développés [127, 207]. La complexité de comprendre le système visuel humain est due à la complexité du cerveau constitué de dizaines de milliards de cellules avec des interconnexions compliquées.
- 2) Le désir d'obtenir une plus grande quantité d'information à partir d'un plus petit nombre de donnée. En effet on désire obtenir les informations 3D des objets et de la scène à partir d'un petit ensemble d'images 2D (ces dernières constituent elles même un problème non encore complètement résolu notamment le problème du bruit) [91]. L'invariance du système visuel humain à la rotation, aux dimensions, et aux positions des objets est un autre problème que les systèmes de reconnaissance essayent de résoudre.

1.3.3 Reconnaissance d'objets 3D: Représentation et mise en correspondance

Nous décrivons dans ce paragraphe, les méthodes de représentation d'objets et les algorithmes d'appariement (mise en correspondance) utilisés par les systèmes de reconnaissance.

Comme c'est mentionné précédemment, un système de reconnaissance est constitué de deux étapes principales : l'étape de représentation (hors ligne) et l'étape de reconnaissance (en ligne). L'étape de reconnaissance est composée à son tour de deux étapes: l'appariement et l'estimation de la position [163, 177, 179, 215].

La représentation et la mise en correspondance sont deux étapes cruciales [29, 76, 111, 122, 214, 218]. La représentation permet de réduire la complexité des données 3D en des représentations plus compactes qui peuvent être facilement et efficacement stockées dans la base de modèles. Cependant il faut s'assurer que suffisamment d'informations sont stockées pour pallier aux problèmes du bruit et des objets en occlusion. Ceci est laissé pour l'étape de mise en correspondance qui emploie généralement des techniques permettant de résoudre un tel problème [14, 32, 38, 52, 71, 167, 108, 164, 21, 26, 41].

Il existe deux principales catégories de méthodes pour la représentation d'objets [136, 138]. Ce sont la représentation basée objet (object-centred representation) et la représentation basée apparence (view-centred representation).

La représentation basée objet décrit les volumes de l'espace occupés par ces objets [142], autrement dit, elle décrit les objets en termes de modèles 3D.

La représentation basée apparence [35, 125, 158] exploite les apparences des objets prises à partir de plusieurs points de vues dans le but d'arriver à une reconnaissance subséquente de l'objet (figure 1.2).

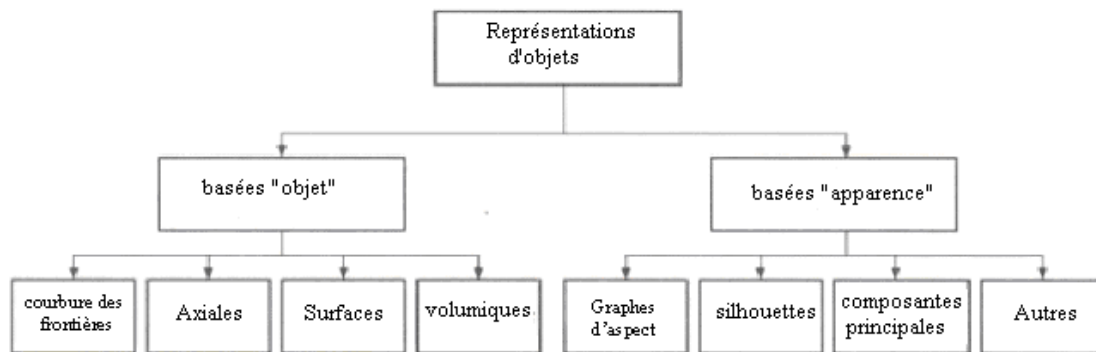


Figure 1.2 Représentations d'objets pour la reconnaissance

Les méthodes de ces quinze dernières années se consacrent pour la reconnaissance des objets à forme libre initialement définis par Besl [30]. Ces méthodes s'éloignent des restrictions imposées par les méthodes antérieures qui modélisent les objets sous forme polyédrique. Cependant d'autres recherches actuelles combinent les techniques polyédriques avec des méthodes statistiques pour la représentation d'objets [123].

1.3.3.1 Représentations basées Objets

Les représentations basées objets peuvent être décomposées en quatre catégories principales:

- Les représentations basées sur les courbures des frontières.
- Les descriptions axiales.
- Les descriptions surfaciques.
- Les descriptions volumiques.

1.3.3.1.1 Représentations par courbure des frontières

Les représentations par courbure des frontières stockent le modèle de l'objet comme étant un triplet [32] constitué de :

- Un ensemble de surfaces.
- Un ensemble de courbes représentant les intersections des surfaces
- Un graphe qui décrit la connectivité des surfaces et contient d'autres informations concernant ces dernières.

La représentation SCS (surface-côté-sommet) est parmi les plus vieilles représentations [32], elle était utilisée pour les objets constitués de surfaces planes et de côtés droits [38,91].

Banau et Faugeras [33] utilisent un étiquetage stochastique et hiérarchique dans le but de résoudre le problème d'appariement de segments. Une partie d'une forme d'un objet de la scène est mise en correspondance avec le maximum de représentations. Les formes considérées pour la reconnaissance sont modélisées par des polygones, permettant des transformations rigides. Des caractéristiques telles que les angles et les longueurs des segments sont extraites des représentations et la probabilité de ressemblance entre les segments de l'objet et ceux du modèle est ensuite calculée.

Ponce et Brady [165] effectuent un lissage gaussien sur les images à différentes échelles et utilisent plusieurs types de primitives pour la représentation d'objets.

Les invariants sont utilisés dans la reconnaissance [72, 152, 172]. Ce sont des propriétés des configurations géométriques qui restent inchangées à la suite des transformations appropriées. Par exemple l'intersection et la colinéarité sont préservées à la suite d'une transformation perspective.

Zisserman et Al. [219] présentent un système de reconnaissance où les bords extraits des images sont utilisés pour calculer trois types d'invariants projectifs à partir d'une conique, de deux droites et d'une paire de coniques. Les invariants sont ensuite indexés dans une table de hachage et utilisés pour créer des classes d'objets vérifiant les mêmes critères d'invariance.

Weiss et Ray [208] adressent le problème de trouver une correspondance de caractéristiques dans le cas de perte de l'information de profondeur lors de passage d'un objet 3D à une simple image 2D. Ils établissent des relations algébriques entre les invariants des modèles 3D et ceux de leurs images. La reconnaissance est effectuée en utilisant les points 3D appartenant à l'espace de la discontinuité de profondeur. En utilisant les relations 2D-3D obtenues il devient possible de mettre en correspondance les caractéristiques des modèles avec celles de la scène.

Les méthodes les plus populaires utilisant les courbures se basent sur les B-splines, qui fournissent des représentations invariantes à la suite des transformations affines et projectives

ce qui est important pour le processus de reconnaissance. Les splines sont des représentations locales et globales ; d'un côté une fonction décrit la courbe entière mais en même temps elle fournit une contrôlabilité locale car le mouvement du point de contrôle change la forme de la courbe seulement autour du point.

Guenzic et Ayache [89] se basent sur des courbes B-Splines non uniformes pour créer des lignes extraites d'images médicales. Les caractéristiques utilisées pour la reconnaissance des courbes sont les courbures et la torsion calculées analytiquement à partir de la représentation des splines.

Cohen et Wang [54] modélisent les courbes 3D avec un marquage distinctif. Ils estiment le meilleur ordre des splines et le meilleur ordre des points de contrôle à utiliser pour la représentation des contours extraits des surfaces d'objets.

* Critiques des Représentations par courbure des frontières

Généralement le plus grand problème avec les techniques basées sur les frontières des objets est son insuffisance pour la description d'objets 3D et ce à cause de la perte d'informations lors de la réduction d'une surface à une courbe. Beaucoup de techniques basées sur les courbures ne sont pas robustes au bruit ce qui réduit l'efficacité du système [22]. L'utilisation des splines est un processus non linéaire [32,59], quand aux méthodes gaussiennes, elles nécessitent de connaître la taille de la fenêtre pour le filtre ainsi que la variance du bruit.

1.3.3.1.2 Descriptions Axiales

Le cône généralisé (CG) est une représentation célèbre définie par une ligne droite autour de laquelle un cercle de rayon croissant balaye l'espace de haut en bas [154] (figure 1.3).

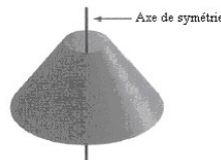


Figure 1.3 Le cône généralisé

Les cônes généralisés sont utilisés dans le système très connu ACRONYM de Brooks [39], et aussi dans les systèmes de reconnaissance par composantes [38].

Das et al [56] ont développé un algorithme employant des modèles hiérarchiques de cônes pour la reconnaissance d'objets dans les scènes d'extérieur. La description des représentations est ensuite complétée en utilisant les frontières convexes et concaves, et aussi des formes polyédriques décrivant des parties de l'objet. Ces représentations multiples d'objets ou plutôt de parties d'objets sont utilisées pour extraire les régions d'intérêt puis leur appariement avec les régions du modèle est effectué.

*** Critiques des Descriptions Axiales**

Les cônes sont excellents pour la représentation des objets ayant une symétrie axiale car la représentation dans ce cas nécessite un petit nombre de paramètres. Cependant pour des formes plus ou moins compliquées il devient très difficile de générer une description conique qui approxime d'une manière assez fidèle le forme initiale, en plus du bruit qui peut affecter ce genre de description, ainsi que les restrictions que présentent cette méthode dans le cas d'environnement contenant des objets occultés [186].

1.3.3.1.3 Descriptions Surfaiques

Initialement les chercheurs en vision artificielle s'inspiraient des méthodes de l'infographie et utilisaient les polyèdres et les surfaces quadratiques pour approximer des portions de segments appartenant aux surfaces d'objets. De telles méthodes étaient incapables d'extraire des caractéristiques fiables, mais avaient plutôt tendance à segmenter la surface en un ensemble de pièces et à relier ces pièces jusqu'à ce que la forme obtenue devienne suffisamment acceptable ce qui génèrait des représentations instables et difficiles à manipuler.

Pour cette raison les chercheurs se sont orientés à extraire d'autres types de primitives et concevoir d'autres techniques de segmentation.

Parmi les techniques connues dans cette catégorie on peut citer l'image gaussienne étendue (EGI) [96] et l'image gaussienne étendue complexe (CEGI) [104]. Leur concept est de représenter l'objet 3D par une collection des normales de surfaces donnant ainsi naissance

à une sphère gaussienne (figure 1.4). Une représentation similaire [153] utilise les tangentes des surfaces d'objets.

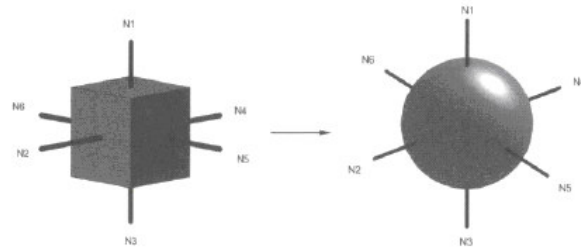


Figure 1.4 Les représentations gaussiennes

La segmentation des surfaces en parties a aussi été un domaine de recherche pour la reconnaissance. Besl et Jain [32] ont élaboré une méthode utilisant le signe (K) et la moyenne (H) de la courbe gaussienne pour segmenter la surface 3D en parties comme appartenant à l'un des huit principaux types indiqués dans la figure 1.5.

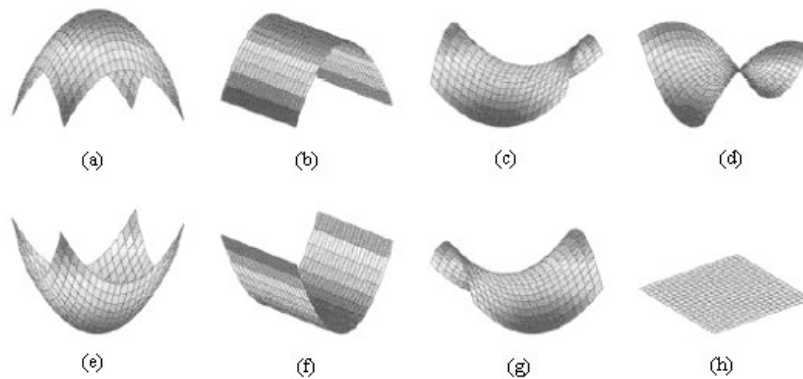


Figure 1.5 Les huit principaux types de surfaces 3D. (a) $H < 0$ et $K > 0$, (b) $H < 0$ et $K = 0$, (c) $H < 0$ et $K < 0$, (d) $H = 0$ et $K < 0$, (e) $H > 0$ et $K > 0$, (f) $H > 0$ et $K = 0$, (g) $H > 0$ et $K < 0$, (h) $H = 0$ et $K = 0$. [23]

Faugeras et Hebert [68] ont développé une méthode de représentation des formes 3D avec une combinaison de points (les sommets des objets ainsi que leurs centres de symétrie) et de lignes (les bordures internes et les axes de symétrie). L'appariement d'un objet avec un modèle est effectué via la méthode de vérification d'hypothèses. D'autres méthodes [67, 102] décomposent la surface en un ensemble de triangles 3D.

La représentation en Splines est également très employée pour la représentation et la reconnaissance d'objets à formes libres [106, 128, 162].

Arman et Aggarawal [14] proposent un système qui à partir de six surfaces types construit une arborescence d'interprétations utilisée pour la reconnaissance. D'autres techniques utilisent des ondelettes et de fonctions sinusoïdales pour la reconnaissance [70, 190].

Un autre axe de recherche dans le cas des méthodes surfaciques qui a vu le jour ces quinze dernières années consiste en l'utilisation de la géométrie différentielle pour la représentation des objets à forme libre. On distingue trois types :

- Les approches globales.
- Les approches locales.
- Les approches hybrides.

Les méthodes différentielles globales utilisent des primitives qui décrivent le volume de l'objet en entier [48,93].

Les méthodes locales n'ont pas besoin de primitives volumiques ou de modélisation de surfaces, les descriptions sont déterminées par des parties restreintes des surfaces d'objets [31, 188, 53]. La plupart de ces méthodes se basent sur la table de hachage pour la reconnaissance.

Vu l'instabilité des représentations basées sur les méthodes globales notamment dans le cas d'occlusion, et la sensibilité au bruit des techniques locales, d'autres chercheurs ont préféré combiner les deux précédentes, en donnant naissance aux approches hybrides [103, 60, 191, 193, 107] dont la plupart utilisent les histogrammes pour l'appariement d'objets avec les modèles.

*** Critiques des Descriptions Surfaciques**

Le problème avec les techniques EGI et CEGI (l'image gaussienne et l'image gaussienne étendue complexe) est le manque d'informations en ce qui concerne les distances explicites ce qui empêche l'estimation de la translation d'objets dans l'espace, elles réduisent aussi la ressemblance des courbes convexes et concaves ayant la même forme. Mais le grand

problème est le fait que des objets différents peuvent avoir la même représentation. Le problème se pose aussi lorsqu'il y a plusieurs objets dans la scène notamment en présence du phénomène d'occlusion [153]. Ce sont des techniques non robustes au bruit et chères en temps de calcul.

La méthode des Splines est très conseillée pour l'infographie mais ne donne pas de bons résultats dans le cas de la vision artificielle. Ceci est dû à l'utilisation des représentations paramétrées pour modéliser des formes libres, ce qui rend difficile son utilisation dans les algorithmes de mise en correspondance de l'objet avec un modèle [162].

1.3.3.1.4 Descriptions Volumiques

Les techniques de représentation volumiques sont essentiellement basées sur :

- Les voxels
- Les octrees
- Les super quadratiques
- Les CSG (constructive solid geometry).

Les voxels sont des primitives volumiques élémentaires (comme les pixels dans le cas 2D) ils ont généralement une forme cubique ou parallélépipède.

Greenspan et Boulanger [81] modélisent chaque objet de la base avec des modèles à base de voxels générés selon différentes positions de l'objet. La reconnaissance est effectuée en comparant un objet donné avec les modèles de voxels stockés en mémoire.

Les octrees sont des versions un peu avancées de la manipulation des voxels. L'octree commence par un nœud représentant une région cubique de la scène, les nœuds sont ensuite étiquetés. Le nœud est dit « rempli » s'il est complètement dans l'objet, il est dit « vide » s'il ne contient aucune partie de l'objet et « partiel » s'il contient une partie de l'objet (Figure 1.6).

La subdivision se fait récursivement jusqu'à arriver à obtenir uniquement des feuilles de type « rempli » ou « vide » [91, 175].

Les supers quadratiques constituent une autre représentation empruntée des techniques de synthèse d'images. Elles décrivent un nombre plus large de volume en ajoutant des

paramètres de déformations tels que l'étirement, les torsions et des contraintes booléennes [90, 161, 199] (Figure 1.7).

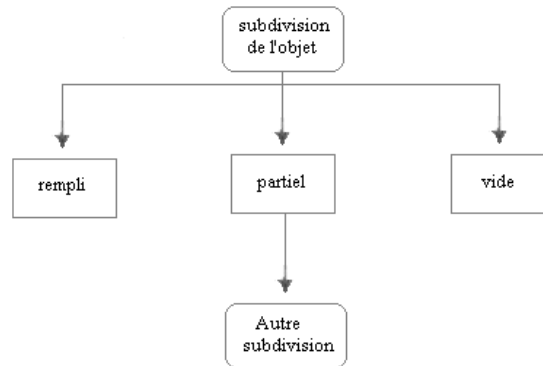


Figure 1.6 La procédure récursive de la représentation en octree

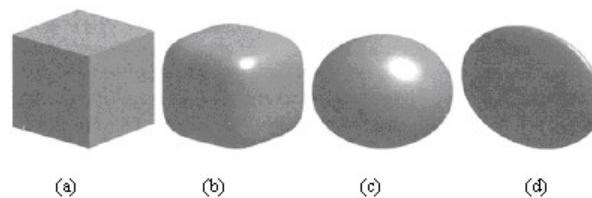


Figure 1.7 L'effet des paramètres de déformation sur les formes 3D

Les méthodes CSG construisent des formes 3D à partir d'une sélection de primitives telles que les cylindres, les sphères et les parallélépipédiques (Figure 1.8). Le modèle est stocké sous forme d'arbre binaire où chaque feuille représente une primitive volumique et chaque nœud interne représente les opérations booléennes effectuées sur les primitives de base (sphères, cylindres ...) [32, 91, 180].

* Critiques des Descriptions Volumiques

La technique des voxels consomme beaucoup de mémoire comme toutes les méthodes décrivant les objets par de très petites primitives, cependant leurs algorithmes sont très faciles à créer et à utiliser [32, 91].

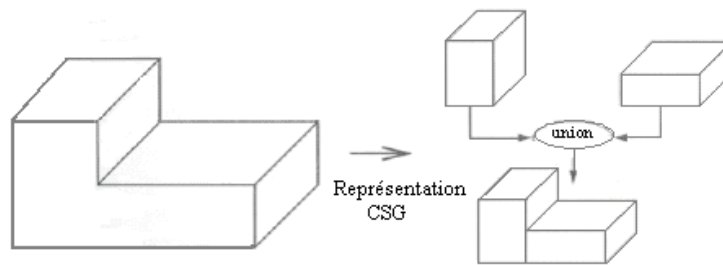


Figure 1.8 La représentation CSG d'un objet simple

Le problème avec les techniques des octrees se manifeste lorsqu'elles sont appliquées sur des objets courbés très complexes, dans ce cas la subdivision récursive du volume devient insuffisante si on veut obtenir une approximation raisonnable concernant la forme de l'objet.

L'inconvénient des méthodes supers quadratiques est dû à leur symétrie dans la représentation, ce qui réduit considérablement la classe d'objets qu'elles puissent représenter.

Le principal problème avec les méthodes CSG réside dans l'ambiguïté de reconstruction d'une forme à partir d'autres formes plus élémentaires. Pour cette raison, on préfère en général, les représentations surfaciques à celles volumiques.

1.3.3.2 Représentations basées Apparence

Les représentations orientées apparence se sont inspirées de la vision humaine. Les objets sont représentés par le système visuel humain par un ensemble de vues ou d'aspects 2D au lieu d'une simple vue 3D [59, 25]. Pour cela les techniques orientées apparence tentent de représenter un objet 3D par tous les aspects 2D possibles. Nous distinguons trois principales représentations utilisées dans le contexte de la reconnaissance :

- Les graphes d'aspect.
- Les silhouettes.
- L'analyse des composantes principales.

Nous présenterons en plus quelques autres méthodes qu'on ne peut pas classer exclusivement dans l'une de ces trois catégories.

1.3.3.2.1 Les graphes d'Aspect

Koenderink et Doorn [112] ont proposé l'utilisation des graphes d'aspects. Il s'agit d'une structure de graphes dans laquelle les aspects 2D d'un objet 3D sont collectés dans des groupes distincts appelés « aspects ». Les nœuds du graphe représentent les différents aspects et les arcs représentent les relations entre les aspects et le passage d'une vue à l'autre.

Un algorithme populaire pour construire le graphe d'aspects, tisse une sphère imaginaire autour de l'objet et calcule toutes les vues possibles, puis regroupe ces dernières selon leur ressemblance topologique. L'utilisation des graphes est reprise dans les travaux actuels et font parties des chapitres de récents ouvrages sur l'indexation, la recherche et la reconnaissance d'images [212, 211].

*** Critiques des Descriptions utilisant le graphe d'aspects**

Malgré la nature intuitive de cette représentation, la taille énorme pour le stockage des données ainsi que la complexité d'une modélisation sous forme de graphes même dans le cas d'objets simples a entravé l'utilisation d'une telle technique dans le cadre de la reconnaissance.

1.3.3.2.2 Les Silhouettes

Dans le cas où l'objet est dépourvu des marques particulières de surfaces telles que les textures, on peut utiliser les silhouettes d'objets comme primitives d'images pour la représentation et la reconnaissance. Il s'agit en fait de décrire la forme d'une surface à travers le contour de celle-ci.

Ullman et Basri [202] utilisent une technique où les projections des frontières d'objets entre les aspects peuvent être calculées à partir des correspondances entre les points de trois aspects seulement. Vaillant et Faugeras [203] calculent les contours d'objets en se basant sur le rayon de courbure d'une section plane et en utilisant aussi des propriétés différentielles.

Chien et Aggarawal [51] reconnaissent des objets 3D à partir d'images 2D en se basant sur les représentations en quadree pour les aspects 2D et la représentation en octree pour l'aspect 3D. En utilisant des intersections volumiques les auteurs génèrent des représentations en octree à partir des principales représentations en quadree extraites à partir de trois vues non

coplanaires. Cette représentation permet la reconstruction des données 2D et aussi 3D à partir d'un ensemble de silhouettes.

D'autres travaux [27, 204] retrouvent la structure 3D d'un objet courbé ainsi que le mouvement de la caméra à partir des silhouettes extraites d'une séquence d'images, et ce en se basant pour chaque image sur des caractéristiques telles que les bitangentes, les tangentes parallèles et les points d'inflexion. Ces caractéristiques des frontières stéréo permettent d'évaluer les paramètres des capteurs de prise de vues. Ensuite en utilisant le mouvement des caméras, les surfaces 3D peuvent être reconstruites.

Mokhtarian [147] a conçu un système capable de reconnaître des objets à forme libre en se basant sur les silhouettes préalablement segmentées pour l'extraction des frontières d'objets. Il effectue une segmentation à plusieurs échelles et à différentes résolutions. Cette représentation commence par l'acquisition d'un modèle de contours reflétant les distances relatives d'un objet vu d'une certaine distance. La courbure le long du contour est utilisée pour construire une table d'indexation utile pour la mise en correspondance.

Ming et Bhanu ont développé le système ORACLE [145]. Ce système reçoit en entrée une image en intensité pouvant contenir des objets non préalablement modélisés. Les objets sont segmentés et les silhouettes sont extraites et représentées par des morceaux de segments de droites. Chaque région est étiquetée puis la vérification d'hypothèses permet de déterminer sa position. Si la reconnaissance échoue, le système tente de reconstruire un autre modèle à partir des primitives extraites. Ces dernières étant regroupées en mémoire sous forme d'arborescence.

Chen et Stockman [49] reconnaissent aussi des objets à forme libre à partir d'une seule image en niveau de gris. Deux types de contours sont utilisés : les silhouettes et les contours internes visibles à l'intérieur de l'aspect.

*** Critiques des Descriptions utilisant les silhouettes**

Ces méthodes sont très intéressantes du moment qu'elles ne consomment pas beaucoup de place mémoire et sont assez rapides en temps d'exécution, ceci s'explique par la représentation d'un objet 3D uniquement par le contour de son aspect 2D. Cependant elles

sont limitées au cas de la reconnaissance d'objets non texturés et surtout lorsque les détails internes n'intéressent pas les domaines de leur application.

1.3.3.2.3 Analyse des composantes principales

Les techniques d'analyse en composantes principales (ACP) ont largement été utilisées en vision artificielle, allant de la reconnaissance des visages [201] jusqu'à la détection des défauts de texture [137]. Cette technique est connue aussi sous le nom de la transformation de Karhunen-Loeve. Elle est basée sur la représentation d'images en termes d'un ensemble compact d'images modèles orthogonales.

Le travaux de Muarse et Nayar [151] et de Leonardis et Bishof [126] sont des exemples des approches ACP pour la reconnaissance d'objets. Pour chaque objet à stocker dans la base d'images, plusieurs vues sont prises sous différentes conditions d'éclairage. Les vues sont normalisées selon la luminance et le facteur d'échelle, puis translatées dans le même espace de coordonnées. Les coordonnées de l'objet 3D sont ensuite reconstruites à partir de toutes les vues 2D stockées.

Ohba et Ikeuchi [157] utilisent les ACP pour la reconnaissance d'objets partiellement occultés où des apparences multiples d'un objet sont stockées puis utilisées pour la reconstruction du modèle 3D.

*** Critiques des Descriptions en composantes principales**

La plupart des méthodes d'analyse en composantes principales nécessitent que les vues soient prises en contrôlant l'éclairage de la scène, et la reconnaissance échoue souvent dans le cas où la source de lumière n'est pas délicatement contrôlée. En plus ces techniques consomment beaucoup de place mémoire car tous les aspects et sous différentes conditions d'éclairages doivent être stockés dans la mémoire du système.

Cependant, Zaeri et AL [216] ont proposé, récemment, une technique basée sur la représentation en composantes principales dans le cadre de la reconnaissance de visages et ce en optimisant l'espace mémoire nécessaire pour la représentation.

1.3.3.2.4 Autres techniques

Edwards et Muarse [64] ont développé une technique basée sur l'apparence, elle consiste à reconstruire une scène à partir d'images présentant des occlusions. Des heuristiques simples ont été utilisées pour la reconnaissance, ils utilisent seulement une seule vue de l'objet dans un environnement à occlusion.

Pope et Lowe [131] utilisent des approches basées sur les distributions de probabilités pour décrire les possibles variations entre les aspects d'un objet. Ils se basent en fait sur trois probabilités concernant respectivement la présence de l'objet dans la scène, sa position ainsi que la valeur donnée pour chaque aspect.

Le système MORAL [115] est un exemple de système utilisant l'apparence pour la navigation autonome. Le système approxime les vues par des polyèdres et utilisent en plus la couleur pour l'étape de la reconnaissance. D'autres méthodes utilisent aussi les ondelettes (wavelet) pour représenter et reconnaître les aspects des objets [210, 28].

Un schéma actuel pour la reconnaissance d'objets utilise l'extraction d'indices localisés. L'extraction de points peut se faire avec des détecteurs de points d'intérêt [133, 205] selon une grille régulière, ou encore avec des points choisis au hasard dans l'image [156]. Typiquement, des descripteurs sont calculés en chacun de ces points permettant d'extraire l'information utile et robuste aux transformations affines et conditions d'illumination. Le plus populaire étant le SIFT (Scale Invariant Feature Transform) [133] bien que suivant les applications on puisse leur préférer par exemple des filtres en ondelettes ou des shape-context [144].

Latecki et AL [124] exploitent une mesure partielle de similarité pour trouver l'objet qui ressemble le plus à une requête donnée, et ceci quand la requête est une partie caractéristique de l'objet à reconnaître. L'inspiration vient du système visuel humain qui est capable de reconnaître un objet en ayant simplement une partie caractéristique qui le représente. Par exemple si on voit une main devant une porte, on sait très bien que c'est une personne qui est derrière la porte.

D'autres méthodes basées sur l'apparence [169, 170] sont appliquées pour détecter le mouvement et les activités des individus dans une séquence d'images vidéo. Les activités sont

reconnues indépendamment de la nature du mouvement et même en cas d'une occlusion partielle. Cette approche consiste à reconstruire un modèle basé sur l'apparence pour chaque individu rencontré dans les séquences. Ce modèle est utilisé par la suite pour l'apprentissage du mouvement en vue de la reconnaissance des individus ainsi que leur mouvement et leurs activités. La stéréovision ainsi que le mouvement non rigide tridimensionnel sont aussi utilisés dans une autre méthode [166] pour minimiser l'erreur quant à la prédiction de la forme et du mouvement, et traite entre autre le problème de l'occlusion et de la distorsion perspective.

1.3.3.3 Récapitulatif de quelques représentations

Nous avons exposé dans les paragraphes précédents deux principales techniques de représentation et de reconnaissance d'objets. L'inconvénient majeur des techniques basées objets réside dans leur complexité vu le temps de calcul et l'espace de stockage sans oublier leur sensibilité au bruit. Quant aux approches basées apparence, le souci majeur réside dans la difficulté de se restreindre à un petit nombre de meilleurs aspects représentatifs de l'objet. Cependant cela reste difficile, voire impossible de dire quelle approche ou même quel algorithme est mieux que l'autre, car ceci dépend des domaines d'application et surtout d'un ensemble de contraintes imposées et vérifiées.

Bennamoun et Mamic [23] ont étudié quelques caractéristiques de quelques méthodes vues précédemment et que nous résumons dans le tableau ci-dessous:

Représentation	Globale	Locale	Non ambiguë	Consistante
Cylindre généralisé	Oui	Non	Oui	Oui
Sphères	Oui	Non	Non	Oui
Courbure	Non	Oui	Non	Non
Splines	Oui	Oui	Oui	Oui
Voxels	Non	Oui	Oui	Non
Supers quadratiques	Oui	Non	Oui	Oui
Silhouettes	Oui	Non	Oui	Oui
ACP	Oui	Non	Oui	Oui

Tableau 1.1 Principales caractéristiques des méthodes de représentation d'objets 3D [23].

1.3.3.4 Mise en correspondance (Matching)

Nous présentons dans cette section les six approches de mise en correspondance les plus populaires utilisées dans le cadre de la reconnaissance d'objets. Il s'agit de :

- Hypothèse et vérification
- Structures relationnelles
- Regroupement de positions
- Hachage géométrique
- Arbres d'Interprétation
- Enregistrement 3D et transformations de distance

1.3.3.4.1 Hypothèse et vérification

Le principe de cette approche consiste à trouver un nombre limité de correspondance entre l'image et le modèle puis calculer la transformation 2D ou 3D du modèle par rapport à la scène [91]. La transformation hypothétique est alors évaluée en se basant sur des mesures statistiques appropriées.

Par exemple, on extrait N points de courbure appartenant à une surface de la scène et on met en correspondance l'un de ces points avec un point d'une courbure similaire appartenant à un modèle stocké dans la base de modèles. Ceci nous permet de générer la transformation reliant les deux points. Ce processus est alors répété pour les N-1 points restants. Nous obtenons ainsi d'autres transformations, et dans le cas où les transformations sont similaires alors nous dirons que les points sont appariés ce qui confirme l'hypothèse d'origine. Dans le cas contraire d'autres hypothèses doivent être considérées puis vérifiées.

Les systèmes utilisant cette approche de vérification d'hypothèses se basent sur le théorème de consistance du point de prise de vue proposé par Lowe [131]. Cependant il reste toujours quelques soucis d'implémentation de cette approche :

- 1- Comment choisir les primitives initiales de la scène pour les mettre en correspondance avec le modèle?
- 2- Quel est le nombre optimal de correspondance nécessaire pour établir une bonne hypothèse sur la transformation perspective ?

3- Comment doit-on évaluer la correspondance entre l'objet et le modèle ?

En plus de ce qui précède, la complexité algorithmique dépend fortement du nombre de modèles stockés.

Uuttenlocher et Ullman [100] utilisent cette technique pour la reconnaissance d'objets solides. La représentation est basée sur les points d'inflexion des contours. En utilisant ces points, ils calculent la transformation 2D-3D entre le modèle et l'image, et ceci en utilisant les points de l'image et trois paires de points du modèle. Le système de reconnaissance utilise cette transformation pour déterminer les alignements possibles puis les hypothèses sont vérifiées en comparant les contours du modèle avec ceux de l'image.

1.3.3.4.2 Structures relationnelles

La forme préliminaire des structures relationnelles utilisées exclusivement dans les représentations à base de contours a été connue sous le nom ARG : attributed relational graph, (les graphes de relation attribués). L'ARG est un ensemble de nœuds correspondant à des primitives, ces nœuds sont reliés par des arcs représentant les relations entre les primitives des nœuds adjacents [59]. L'appariement entre le graphe d'un modèle et celui d'une scène est effectué en utilisant les techniques de la théorie des graphes et aussi l'isomorphisme de sous graphes. Le problème majeur avec cette approche est la complexité NP complet des algorithmes, signifiant que le temps d'appariement croît d'une manière exponentielle en augmentant le nombre de nœuds [91]. Une nouvelle génération des ARG est l'hyper ARG (Attributed hyper graph). Chaque sommet correspond à une surface de l'objet, et un ensemble de sommets représentent des primitives appartenant à plusieurs blocs de l'objet (Figure 1.9). C'est une représentation qui permet la réduction du temps de mise en correspondance mais malheureusement les algorithmes demeurent toujours NP complets [209].

Shapiro [181,182] présente une technique symbolique et numérique qui peut être employée pour appairer des descriptions structurelles d'un objet et d'un ensemble de ses modèles. En ayant deux modèles, il définit leur distance relationnelle et utilise cette distance pour déterminer le modèle le plus proche de l'objet. Il se base surtout sur les techniques d'histogramme pour éliminer les mauvais candidats puis effectue un partitionnement (une classification) des bases de modèles.

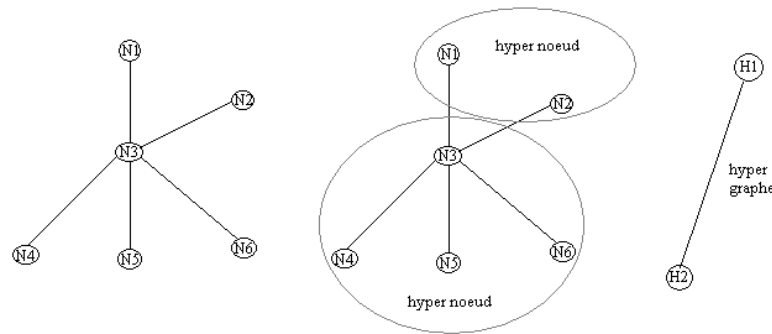


Figure 1.9 Exemple de représentation en ARG et hyper ARG

Le regroupement de modèles et les descriptions symboliques peuvent être utilisés pour un meilleur partitionnement et une bonne recherche dans la base de modèles. Les modèles peuvent être regroupés en classes, chaque classe est représentée par un seul modèle. Un objet requête va par la suite être comparé au représentant du groupe au lieu d'être comparé à tous les modèles de ce groupe.

1.3.3.4.3 Regroupement des positions

L'approche utilise K appariements de caractéristiques (qui contraignent la position de l'objet) entre le modèle et l'image. Le regroupement de positions pour la reconnaissance [189] implique la formation de toutes les correspondances entre les K caractéristiques du modèle avec K caractéristiques de l'image. Des paramètres qui décrivent la transformation sont établis pour chaque point. Cette dernière sera ensuite soumise à une étape de vote. Le même processus est refait pour d'autres positions de l'objet.

La transformation de Hough est une technique connue, cependant Grimson et Huttenlocher [84] ont analysé cette transformation et trouvé qu'elle convient peu pour la reconnaissance d'objets dans des environnements complexes.

Ce type de mise en correspondance dépend du nombre d'objets de la base, du moment qu'elle nécessite d'apparier K primitives entre l'image et le modèle, ajouté à cela les opérations de vote nécessitant une très grande place de stockage car on prend en considération toutes les positions de l'objet pour l'étape de vote.

1.3.3.4.4 Le hachage géométrique

Le hachage géométrique est une méthode de stockage, de mise en correspondance et de recherche de l'information dans une table de hachage, utilisant les index entiers issus d'une fonction de hachage à partir des clés définies par les invariants. Le principe du hachage est de calculer l'adresse entière nommée index, de l'emplacement d'un élément dans la table de hachage à partir de sa clé, définie par une fonction de hachage. Carrez [42] a démontré que la recherche est d'autant plus optimale quand la fonction de hachage tend vers la fonction uniforme [59, 113, 171].

LAMDAN et WOLFSON [113] présentent une application du hachage géométrique dans le domaine de la reconnaissance. Les objets sont modélisés par une représentation des ensembles de points d'intérêt 2D, invariants à une transformation affine.

L'étape de l'apprentissage (pré-traitement) consiste à extraire les coordonnées de N points d'intérêt (le critère d'extraction n'est pas important). Un triplet de points non colinéaires est choisi pour former une base affine du plan. Les coordonnées des $N-3$ autres points d'intérêt sont considérées comme invariantes. Dans un second temps, trois autres points non colinéaires sont pris pour former une nouvelle base, de façon à définir les descriptions invariantes de $N-3$ points restants ; et ainsi de suite...

La méthode génère toutes les combinaisons possibles de trois points, parmi N . Ces combinaisons sont ensuite caractérisées par leurs coordonnées exprimées dans les repères affines. Ces coordonnées constituent les clés d'indexation. Une fonction de hachage à deux dimensions est mise en place, recevant pour tout point caractéristique une référence au modèle auquel il appartient, ainsi que l'indication du repère affine.

Pour la reconnaissance d'objets, une table de vote est établie, et le même opérateur de points d'intérêt est appliqué pour l'extraction des points d'intérêt d'une image test. Un triplet arbitraire, non colinéaire, est employé comme base affine et les coordonnées des autres points sont utilisées pour voter pour les objets (ou plus précisément pour un triplet particulier et son modèle d'objet). Pour chaque caractérisation, la base d'indexation est consultée et un modèle voit son compteur incrémenté chaque fois qu'une de ses références est trouvée dans la table. Si aucun objet n'a obtenu un nombre suffisant de votes après une étape, il est possible de

choisir un autre triplet non colinéaire pour une autre étape de vote. Si un modèle d'objet a accumulé un nombre suffisant de votes, il peut être vérifié par une mise en correspondance de l'image test et du modèle de l'objet.

Avec cette technique, c'est important d'être prudent quand aux types de primitives à extraire et à utiliser pour l'indexation, car les mêmes primitives peuvent référencer plusieurs objets, ceci afin que la mise en correspondance ne dépende pas de nombres de modèles stockés.

1.3.3.4.5 L'arbre d'interprétation

Un arbre d'interprétation peut être créé de la manière suivante : en utilisant un ensemble de primitives en entrée (des données de l'objet), le premier niveau de l'arbre peut être créé en associant la première primitive modèle (m1) avec toutes les primitives de l'objet (r1 jusqu'à r8) (figure 1.10).

Un deuxième niveau est créé en associant chaque assignement de l'étape précédente avec la deuxième primitive du modèle et ainsi de suite jusqu'à effectuer tous les assignements entre toutes les primitives.

Le but des arbres d'interprétation consiste à trouver les N interprétations consistantes (N est le nombre des primitives modèles à mettre en relation avec les objets) [82, 83, 85].

Le temps de calcul de cette technique est raisonnable seulement lorsque les données appartiennent au même objet. Dans le cas contraire la recherche est exponentielle. Pour cette raison plusieurs contraintes doivent être utilisées pour résoudre ce problème comme les heuristiques par exemple.

1.3.3.4.6 Enregistrement rigide 3D

On entend par enregistrement rigide 3D le processus qui consiste à calculer la transformation rigide tridimensionnelle entre deux données 3D ou plus. Une transformation rigide préserve la taille et la forme des données, elle est composée seulement des rotations et des translations. Les données 3D peuvent être un nuage aléatoire de points ou n'importe quels autres types de données telles que les surfaces implicites ou même paramétrées [47, 98, 99, 183]. Le but d'enregistrer deux ensembles de données ou plus est leur alignement dans le

même système de coordonnées ce qui permettra le calcul direct des relations spatiales entre ces points. Du point de vue de la transformation perspective, le problème d'enregistrement consiste à la détermination d'une matrice 3D de rotation (R), et d'un vecteur (T) de translation permettant un meilleur alignement entre deux ensemble de données X et Y, ce qui peut être algébriquement défini par : $X = RY + T$. Le meilleur alignement correspond souvent à la minimisation de distances euclidienne. Les chercheurs évitent de plus en plus cette approche de mise en correspondance car ça nécessite d'enregistrer la scène 3D pour chaque objet présent dans la base de donnée.

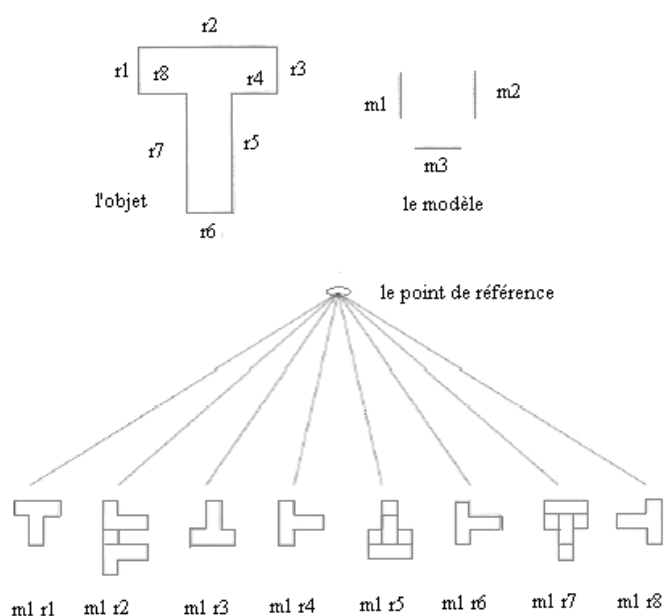


Figure 1.10 Arbre d'interprétation

1.4 Conclusion

La complexité de la vision par ordinateur réside dans sa modélisation en tant que processus informatique. En effet il faut comprendre ce qui doit être calculé à partir des images obtenues et comment réaliser et implémenter efficacement les calculs, puis comment structurer les données 3D afin de localiser et reconnaître les objets.

La vision par ordinateur a suscité un grand intérêt pour les chercheurs depuis plusieurs années. Depuis, le domaine de l'intelligence artificielle ne cesse d'évoluer.

Nous avons présenté dans ce chapitre des notions élémentaires sur la vision artificielle, en particulier la vision binoculaire. Ensuite nous sommes passés à l'étude de deux principales approches de représentation et de reconnaissance d'objets 3D : les techniques basées modèles décrivent les objets en termes de modèles 3D. Leur inconvénient majeur est leur complexité en de temps de calcul et d'espace de stockage.

Les techniques basées apparence exploitent les apparences des objets prises à partir de plusieurs points de vues dans le but d'arriver à leur reconnaissance. Leur souci majeur réside dans la difficulté de se restreindre à un petit nombre de meilleurs aspects représentatifs de l'objet.

Le choix de la méthode de mise en correspondance à utiliser pour la reconnaissance d'objets dépend fortement du domaine d'application. Si on a besoin d'un nombre très restreint d'objets à stocker dans la base d'images, alors les techniques des graphes d'aspect peuvent être utilisées. Cependant comme la taille de la base est très souvent importante les variantes du hachage géométriques sont plutôt conseillées.

Parmi les techniques basées apparence qu'on a étudié dans ce chapitre, celles utilisant la silhouette pour la représentation et la reconnaissance. Ces techniques ne consomment pas beaucoup de place mémoire et sont assez rapides en temps d'exécution sauf qu'on ne peut pas les utiliser si on doit s'intéresser aux textures ou d'autres marques particulières sur les surfaces des objets.

Nous proposons dans le chapitre suivant un état de l'art sur la représentation des objets par leurs silhouettes.

Chapitre 2

Etat de l'art sur les techniques de représentation des silhouettes d'objets

2.1 Introduction

La silhouette est un dispositif visuel important pour la description du contenu des images. Cependant, la description de la silhouette de l'objet est une tâche difficile puisqu'il est laborieux de définir les caractéristiques perceptuelles des silhouettes et de mesurer la similitude entre elles.

La reconnaissance fondée sur la silhouette de l'objet a été un secteur très actif au cours de ces dernières années. La fiabilité, la vitesse et le taux de reconnaissance élevé en constituent les intérêts majeurs de ce concept.

Dans de nombreuses applications de traitement d'images, la reconnaissance des divers objets se fait à partir de leurs silhouettes. C'est particulièrement vrai dans les applications de tri dans lesquelles on est intéressé à saisir ou trier des objets, dans les applications de reconnaissance de caractères, ainsi que pour des applications de surveillance ou de guidage, par exemple en imagerie militaire.

La tâche de perception et la reconnaissance d'objets à partir de leurs images nécessitent de disposer d'une base d'images modèles. Vu le nombre important d'objets de notre monde, ainsi que le nombre d'images différentes pour chaque objet, il sera très coûteux de mémoriser

toutes les images possibles, ce qui impliquera des algorithmes de recherche d'images très coûteux en temps de calcul. Le recours à la description d'images est une solution à ce problème à condition que cette description soit fidèle à l'image et quelle soit facilement manipulable par les algorithmes.

Il est donc important que l'utilisateur dispose d'une représentation de la forme vérifiant plusieurs propriétés :

- Une bonne fidélité à la forme initiale,
- Une bonne discrimination des formes différentes,
- Une bonne adaptation aux opérations de reconnaissance des formes,
- une insensibilité aux déformations qui sont susceptibles d'entacher l'objet car les images peuvent avoir des anomalies dues au bruit, il est donc nécessaire de mettre au point des méthodes pour l'élimination du bruit afin de rendre le processus de reconnaissance possible et non ambigu.
- Une certaine compacité pour permettre l'archivage de nombreuses formes et potentiellement de ces mêmes formes sous divers aspects.

Nous présentons dans ce chapitre un état de l'art sur la représentation d'objets par leurs silhouettes, La dernière méthode présentée concernera une description textuelle sur laquelle se base notre travail.

2.2 Extraction des silhouettes

2.2.1 Représentation d'une silhouette

La représentation d'une silhouette dans une image est une représentation par une fonction caractéristique sous forme d'une image binaire, les seuls pixels non nuls étant ceux couverts par la forme :

$f(i,j)=0$ si le pixel (i,j) n'appartient pas à l'objet.

$f(i,j)=1$ si le pixel (i,j) appartient à l'objet.

Dans une image où de nombreux objets existent, on utilise une image d'étiquettes (ou de labels) qui généralise la notion de fonction caractéristique :

$$f(i,j)=k \quad \text{si le pixel } (i,j) \text{ appartient à l'objet } k.$$

Pour toute représentation de silhouettes, il faut d'abord procéder à leur extraction d'une manière exacte (figure 2.1), mais avant de le faire il faut détecter le contour puis les points d'intérêt de l'objet. Les contours représentent la frontière entre les objets de l'image, ou la limite entre deux pixels dont les niveaux de gris représentent une différence significative.

La détection du contour repose sur l'étude des dérivées de la fonction d'intensité dans l'image, de même les points d'intérêt vont se former lors du croisement des contours.

Pour l'extraction de la silhouette dans l'image, plusieurs techniques peuvent être utilisées. A titre d'exemple, la méthode de seuillage et la méthode de soustraction d'images.

2.2.2 Technique de traitement d'image

L'opération dite de "seuillage" est un procédé de conversion d'une image, elle consiste à mettre à zéro tous les pixels ayant un niveau de gris inférieur à une certaine valeur appelée seuil, et à la valeur maximale les pixels ayant une valeur supérieure. Ainsi le résultat du seuillage est une image binaire contenant des pixels noirs et blancs, c'est la raison pour laquelle le terme de binarisation est parfois employé. Le seuillage permet de mettre en évidence des formes ou des objets dans une image. Toutefois la difficulté réside dans le choix du seuil à adopter.

2.2.3 Technique de soustraction d'images

On utilise la technique de soustraction d'image dans le cas où la scène est observée au moyen d'une caméra fixe. Le procédé de soustraction d'images entre image de fond et l'image courante est utilisé dans le cas d'application de détection d'intrus dans une scène.



Figure 2.1 Exemple d'extraction de silhouette.

2.3 Classification des techniques de représentation des silhouettes

La variété de techniques de représentation de silhouettes en littérature est classifiée (classification hiérarchique) en deux méthodes [217]:

- Les méthodes basées sur le contour où la caractéristique visuelle de la silhouette de l'objet est extraite uniquement du contour.
- Les méthodes basées sur la région où la caractéristique visuelle de la silhouette de l'objet est extraite à partir de la région entière.

Chaque méthode est classée à son tour soit en globale, soit en structurelle.

Les approches globales (continues), ne divisent pas la silhouette en parties secondaires, un vecteur de caractéristiques dérivé de la frontière intégrale de la silhouette de l'objet est employé pour décrire la silhouette.

Les approches structurelles, décomposent la silhouette en parties secondaires appelées « primitives ». Les méthodes structurelles diffèrent dans le choix et l'organisation des primitives pour la représentation de silhouettes.

Les techniques de représentation et description des silhouettes sont regroupées dans la figure 2.2.

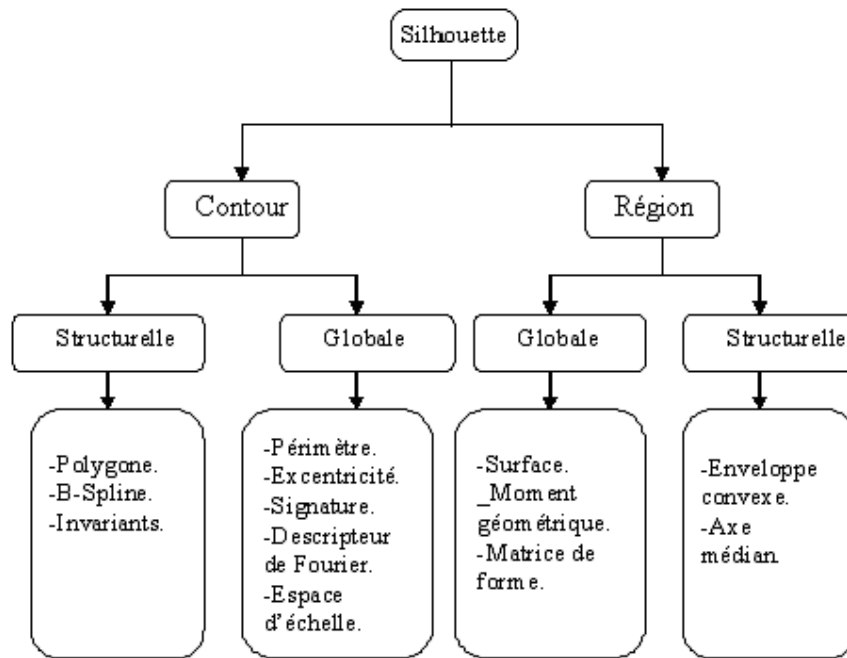


Figure 2.2 Classification des techniques de description et représentation de silhouettes

2.3.1 Les techniques de représentation de silhouettes basées sur les contours

Ces techniques de représentation exploitent seulement l'information de la frontière de la silhouette [217].

Il y a généralement deux types d'approches très différentes pour la modélisation : l'*approche continue* ou globale et l'*approche discrète* ou structurelle. Les approches continues ne partagent pas la silhouette en sous-parties. La mesure de similarité des silhouettes est soit basée sur les points, soit sur les caractéristiques. Les approches discrètes découpent la frontière de la silhouette en segments. La représentation finale est le plus souvent une chaîne de caractères ou un graphe (ou un arbre), la mesure de similarité est faite en associant les chaînes ou les graphes. Dans ce qui suit, nous donnerons un peu plus de détails sur ces approches.

2.3.1.1 Méthodes globales

Les techniques globales de représentation de contours calculent habituellement un vecteur numérique multidimensionnel des caractéristiques de l'information du contour de la silhouette. L'appariement entre les silhouettes est un processus qui utilise la distance métrique telle que la distance euclidienne [217].

Dans cette approche on peut trouver des méthodes comme : les descripteurs de silhouette simple, la signature de silhouette, les moments de frontière, l'appariement élastique, la méthode stochastique, la transformation spectrale...

2.3.1.1.1 Les descripteurs simples de silhouettes

Les descripteurs simples de silhouettes sont des aires, des circulaires, des excentriques (la longueur de l'axe principal / la longueur de l'axe mineur), et aussi celles utilisant l'orientation de l'axe principal [213]. Ces descripteurs globaux simples peuvent seulement distinguer les silhouettes avec de grandes dissimilitudes donc ils sont habituellement utilisés comme filtres pour éliminer les silhouettes qui sont globalement différentes, ou ils sont combinés avec d'autres descripteurs de silhouettes pour les distinguer.

Par exemple, l'excentricité de la forme sur le schéma 2.3.a est proche de 1, elle ne décrit pas correctement la silhouette, dans ce cas-ci, la circularité est un meilleur descripteur. Les segments a et b des deux silhouette sur le schéma 2.3.b et 2.3.c ont la même circularité, cependant, elles sont des silhouettes très différentes. Dans ce cas-ci, l'excentricité est un meilleur descripteur [217].

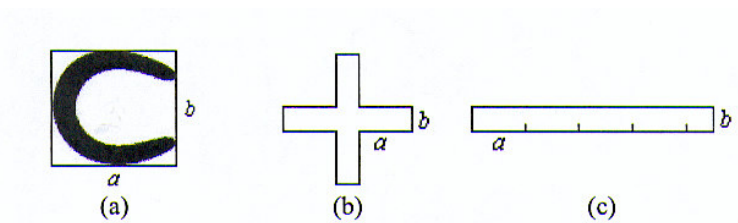


Figure 2.3 Excentricité et circularité des silhouettes

2.3.1.1.2 Les signatures de silhouettes

Les signatures de silhouettes représentent la silhouette par une fonction unidimensionnelle dérivée des points de frontière de la silhouette de l'objet.

Beaucoup de signatures de silhouettes existent, y compris les coordonnées complexes, les coordonnées polaires, distance centrale, angle de tangente, courbure, secteur et la longueur de la corde. Des signatures de silhouettes ont été intensivement étudiées dans [58, 159]. Ces signatures sont habituellement normalisées et invariables, et réduisent l'identification de silhouettes de l'espace 2D à l'espace 1D. Cependant et afin de compenser des changements d'orientation, l'appariement de décalage est nécessaire pour trouver la meilleure similitude entre deux silhouettes d'objets. Alternativement, la signature peut être quantifiée dans un histogramme de signature et employée pour l'appariement des silhouettes d'objets [187].

2.3.1.1.3 Appariement élastique

Dans cette approche, le modèle déformé est généré avec la somme du modèle original et la déformation arrondie.

Cette approche n'est pas pratique pour la récupération en ligne d'images, essentiellement à cause du calcul et de la complexité de l'appariement. Les auteurs [58] ont comparé la complexité du calcul des caractéristiques extraites avec les méthodes dans [155] et [94]. Ils ont démontré que le nombre des opérations du CPU est inférieur, cependant, le nombre d'étapes du processus de déformation est nécessaire pour accomplir l'appariement, ce qui rend l'appariement extrêmement cher.

2.3.1.1.4 La méthode stochastique

Les modèles de séries temporelles en particulier le modèle autorégressif (AR) ont été utilisés pour le calcul des descripteurs de silhouettes [105, 46, 61, 65, 57, 92, 178].

Les méthodes de cette classe sont basées sur la modélisation stochastique 1-D d'une fonction obtenue à partir de la silhouette.

Un modèle autorégressif linéaire exprime une valeur d'une fonction comme une combinaison linéaire d'un certain nombre des valeurs précédentes.

Spécifiquement, chaque valeur de la fonction dans la séquence a une certaine corrélation avec les valeurs précédentes de la fonction et peut donc être prévue par un certain nombre M d'observations des valeurs précédentes de la fonction.

L'inconvénient de la méthode du modèle autorégressif est que dans le cas où la frontière est complexe, un petit nombre de paramètres ne suffit pas pour la description.

2.3.1.1.5 La méthode de l'espace d'échelle

Le problème de la sensibilité au bruit et des variations de frontières dans la plupart des méthodes spatiales trouve une issue dans l'analyse de l'espace d'échelle. Sa représentation est créée en poursuivant la position des points d'inflexion de la frontière de la silhouette filtrée par les filtres gaussiens passe-bas.

Les points d'inflexion qui restent dans la représentation sont prévus pour qu'ils soient des caractéristiques « significatives » de l'objet.

Le résultat est habituellement un arbre d'intervalle, appelé « l'empreinte digitale », constituée par les points d'inflexion. La difficulté avec ce type d'approche est l'interprétation du résultat final.

Le premier essai de ASADA et BRADY était d'interpréter l'arbre d'intervalle acquis de l'espace d'échelle [15]. Leurs arbres d'intervalle sont acquis du filtre gaussien et de sa dérivée seconde.

Mokhtarian et Mackworth [149] adoptent la méthode d'interprétation d'Asada et de Brady. Ils appellent la signature acquise de l'espace d'échelle par « l'espace d'échelle de courbure » (CSS) du contour de l'image ou curvature scale space.

Daoudi et Matusiak ont interprété l'arbre d'intervalle de l'espace d'échelle comme une topologie géodésique [55]. Ainsi, l'appariement entre deux silhouettes revient à apparier les espaces d'échelles respectives des deux images en utilisant la mesure de distance géodésique proposée par Eberly [63].

2.3.1.1.6 La transformée spectrale

Les descripteurs spectraux surmontent le problème de la sensibilité au bruit et aux variations de frontière (courbure) en analysant la silhouette dans le domaine spectral [217].

Les descripteurs spectraux incluent les descripteurs de Fourier (FD) et les descripteurs de Wavelet (WD), ils sont habituellement dérivés de la transformée spectrale des signatures 1_D de silhouettes.

2.3.1.2 Méthodes structurelles

Avec les approches structurelles, les silhouettes sont découpées en des segments de frontière appelés « segments primitifs ». Les méthodes structurelles se distinguent dans la sélection et l'organisation des primitives dans la représentation de la silhouette. La plus commune est basée sur l'approximation polygonale, la décomposition et l'adaptation des courbes [160]. Le résultat est codé par une chaîne de caractères de forme générale : $S = S_1, S_2, \dots, S_n$ Où : S_i peut être un élément du code de la chaîne, le côté du polygone, etc.

La chaîne peut être directement utilisée pour la description ou comme donnée d'entrée dans un analyseur syntaxique de haut niveau.

2.3.1.2.1 Représentation de chaîne de code

Les bases de cette méthode ont été introduites par Freeman en 1961 [73], et qui permet le codage des configurations géométriques arbitraires.

Les courbes arbitraires sont représentées par une séquence de petits vecteurs de la taille unité et l'ensemble fini des directions possibles, ainsi la méthode est appelée : « la méthode du vecteur unité ».

2.3.1.2.2 Décomposition en polygones

Dans [87] et [88], la frontière de la silhouette est décomposée en des segments de droite avec l'approximation polygonale. Les sommets du polygone sont utilisés comme primitives. La caractéristique de chaque primitive est exprimée en quatre éléments qui sont : l'angle interne, la distance du nœud suivant, et les coordonnées x, y .

Bien évidemment, ces caractéristiques sont invariantes à la translation, à la rotation, et au changement d'échelle.

2.3.1.2.3 Analyse syntaxique

L'idée de cette analyse est la similitude entre la composition d'une scène naturelle et la composition d'un langage. Les expressions sont composées de phrases, les phrases sont composées de mots et les mots sont composés de l'alphabet [74].

Dans la méthode syntaxique, la silhouette est représentée par un ensemble prédéfinie de primitives nommé « livre de code » et les primitives sont nommées « mots de code ».

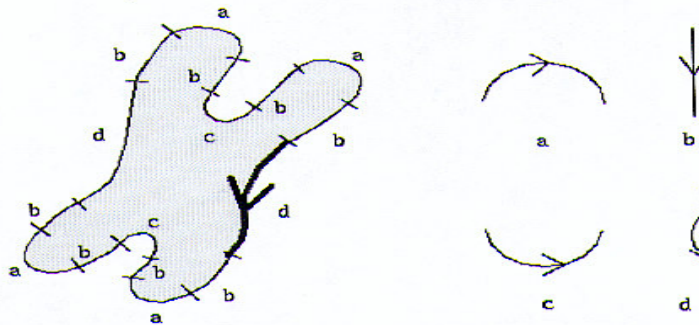


Figure 2.4 Décomposition structurelle d'une silhouette d'un chromosome [186]

Par exemple, les mots de code sont donnés dans le côté droit de la figure 2.4. La silhouette du chromosome (côté gauche de la figure 2.4) peut être représentée comme une chaîne grammaticale par S : $S = d b a b c b a b d b a b c b a b$.

L'appariement entre les silhouettes peut utiliser la chaîne appariée en retrouvant le nombre minimum d'opérations réalisées pour convertir une chaîne à l'autre.

2.3.1.2.4 Les invariants de la silhouette

Cette méthode est vue comme une approche structurelle, car elle présente la silhouette en se basant sur les primitives de ses frontières.

Généralement les théories d'invariants sont basées sur une collection de transformations qui peuvent être composées et inversées

Les invariants les plus connus sont les invariants géométriques, les invariants algébriques et les invariants différentiels.

Les invariants géométriques et les invariants algébriques sont conseillés dans le cas où les frontières peuvent être représentées par des lignes droites ou des courbes algébriques, dans le cas contraire, on utilise les invariants différentiels [217].

2.3.2 Les techniques de représentation de silhouettes basées sur les régions

Dans les méthodes basées sur les régions, tous les pixels de la région sont pris en considération afin d'obtenir la représentation de la silhouette. Les méthodes basées sur la région utilisent les descripteurs de moments pour décrire la silhouette.

D'autres approches incluent la méthode de la grille, la matrice de silhouette, l'enveloppe convexe et l'axe médian.

Comme la méthode de représentation des silhouettes par contour, les méthodes basées sur la région sont aussi subdivisées en globales et structurelles.

2.3.2.1 Les méthodes globales

L'approche globale inclut : les moments géométriques invariants, les moments algébriques invariants, les moments orthogonaux et autres, les méthodes basées sur les grilles et les matrices de silhouettes. Les deux dernières méthodes ne reflètent pas la distribution statistique de la région de la silhouette comme celles basées sur les moments. Elles sont plus sensibles au bruit et sont moins précises que les moments invariants.

2.3.2.1.1 Moments géométriques invariants

Historiquement, Hu a publié le premier papier significatif sur l'utilisation des moments invariants d'images pour les applications bidimensionnelles de reconnaissance de modèles [97]. Son approche est basée sur le travail des mathématiciens du 19^{ème} siècle Boole, Cayley et Sylvester, et la théorie des formes algébriques.

Il utilise des combinaisons non linéaires des moments d'ordre inférieur, d'un ensemble de moments invariants (habituellement appelés moments géométriques), qui ont les propriétés de l'invariabilité en translation, rotation et changement d'échelle. Les moments géométriques invariants ont attiré une grande attention [197, 198, 77, 129, 186, 101] et ils ont été utilisés dans plusieurs applications [62, 20, 168, 146].

Le problème majeur avec les moments géométriques est le fait qu'un petit ensemble d'invariants dérivés de l'ordre inférieur des moments est insuffisant pour décrire exactement la silhouette [143, 174].

2.3.2.1.2 Moments algébriques invariants

Les moments algébriques invariants ont été introduit par Taubin et Tonnelier [194, 195], et ont été employés dans le système QBIC [155, 176]. Les moments algébriques invariants sont calculés à partir des premiers moments centrés et sont donnés comme des valeurs prédéfinies des matrices, dont les éléments sont des facteurs des moments centrés.

Les moments algébriques invariants peuvent être calculés jusqu'à un ordre arbitraire, cependant, les résultats de [176] montrent que les moments algébriques invariants s'exécutent très bien ou très mal sur chacune des requêtes d'objets. Ils tendent à bien fonctionner sur des objets où la distribution de pixels n'est pas importante sur le contour de la silhouette. Sur des objets où la configuration du contour de la silhouette est importante, comme dans la différence entre une forme S et une forme de serpent, les moments algébriques invariants sont moins performants (s'exécutent mal).

2.3.2.1.3 Moments descripteurs de silhouettes

Les moments descripteurs de silhouettes, sont habituellement concis, robustes, faciles à calculer et à apparier. L'inconvénient avec beaucoup de méthodes qui utilisent les moments est qu'il est difficile de corrélérer des moments d'ordre supérieur avec des caractéristiques physiques des silhouettes. Parmi les moments descripteurs de silhouettes, les moments de Zernike sont les meilleurs pour décrire les silhouettes, puisqu'ils sont extraits à partir du domaine spectral, ils ont les mêmes propriétés des caractéristiques spectrales qui sont bien assimilées. La description de silhouettes utilisant les moments de Zernike s'avère être très prometteuse [109, 110].

2.3.2.1.4 Méthodes basées sur les descripteurs de grille

Le descripteur de grille de silhouettes est proposé par Lu et Sajjanhar [134] et a été employé dans [44]. Fondamentalement, une grille d'un certain nombre de cellules est étendue sur une silhouette, la grille est alors balayée de gauche à droite et de haut en bas. Le résultat est une image en mode point (bitmap), les cellules couvertes par la silhouette sont assignées à 1, et celles non couvertes par la silhouette sont assignées à 0. La silhouette peut alors être représentée comme un vecteur binaire de caractéristiques.

La distance binaire ou la distance du bloc local est employée pour mesurer la similitude entre deux silhouettes. Afin de s'adapter à la translation, à la rotation et au changement d'échelle de la silhouette, la distance est d'abord normalisée avant le balayage, la silhouette est posée dans un rectangle de taille fixe, décalée jusqu'au sommet gauche du rectangle et tournée de sorte que l'axe principal de la silhouette soit horizontal. La silhouette reflétée et la silhouette renversée doivent être considérés séparément.

L'avantage du descripteur de grille est sa simplicité de représentation, cependant, le calcul est très cher.

2.3.2.1.5 Matrice de la silhouette

Les méthodes normales de silhouettes, utilisent la grille pour acquérir l'information de la silhouette. La représentation de la silhouette dérivée de cette manière ne peut être traduite, alors que la rotation et le changement d'échelle sont invariants. La normalisation supplémentaire est donc exigée. Goshtasby propose l'utilisation de la matrice de silhouette qui est dérivée d'une technique d'échantillonnage circulaire de trame [79].

L'idée est semblable au prélèvement normal de trame. Cependant, plutôt que de recouvrir la grille carrée normale par une image de silhouette, une trame polaire de cercles concentriques et lignes radiales est posée sur le centre de la masse (figure 2.5).

La valeur binaire de la silhouette est prélevée aux intersections des cercles et des lignes radiales. La matrice de la silhouette est formée de sorte que les cercles correspondent aux colonnes de la matrice et les lignes radiales correspondent aux rangées de la matrice.

La représentation de la matrice résultante est invariable à la translation, à la rotation, et au changement d'échelle.

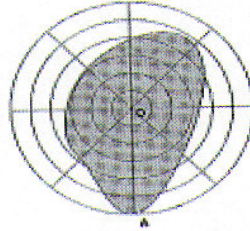


Figure 2.5 Un quadrillage polaire échantillonné d'une silhouette [130]

2.3.2.2 Les méthodes structurelles

Semblables aux méthodes structurelles de contour, les méthodes structurelles basées sur la région, décomposent la région de silhouette en pièces qui sont alors employées pour sa représentation et description. On y trouve ici les enveloppes convexes et les axes médians.

Les méthodes structurelles basées sur les régions ont les mêmes problèmes que celles basées sur les contours. En plus de leurs calculs et implémentation complexes, l'appariement de graphes est un problème qui a besoin lui-même d'être résolu. Il est à noter que toutes les méthodes structurelles décrites ont besoin des informations sur les frontières de la silhouette.

2.3.2.2.1 L'enveloppe convexe

Une région R est convexe si et seulement si deux points quelconque $x_1, x_2 \in R$ alors le segment de droite entier $[x_1x_2]$ est à l'intérieur de la région. L'enveloppe convexe d'une région est la plus petite région convexe H qui satisfait la condition $R \subset H$. La différence $H - R$ s'appelle « l'insuffisance convexe D de la région R . ».

L'extraction de l'enveloppe convexe peut utiliser la méthode de frontières tracées [186] et la méthode morphologique [77, 58]. Les frontières de la silhouette tendent à être irrégulières en raison de la variation de la segmentation, cet effet a habituellement comme conséquence une déficience convexe qui a de petits composants (sans signification) dispersés aléatoirement dans toute la surface. La silhouette peut alors être représentée par une chaîne de concavités.

Une représentation complète de la silhouette peut être obtenue par un processus récursif qui a comme résultat un arbre de concavités. Ici l'enveloppe convexe d'un objet est d'abord obtenue avec ses insuffisances convexes, puis l'enveloppe convexe et les déficiences des déficiences convexes sont trouvées, puis l'enveloppe convexe et les déficiences de ces déficiences convexes et ainsi de suite jusqu'à ce que toutes les insuffisances convexes dérivées soit convexes. La figure 2.6.a illustre ce processus. La silhouette est alors représentée comme un arbre de concavité (La figure 2.6.b). La concavité peut être décrite par ses secteurs, la longueur du pont (La ligne qui relie la coupe de la concavité), la courbure maximum, la distance à partir du point de courbure maximum du pont. L'appariement entre silhouettes devient un appariement de chaînes ou appariement de graphes.

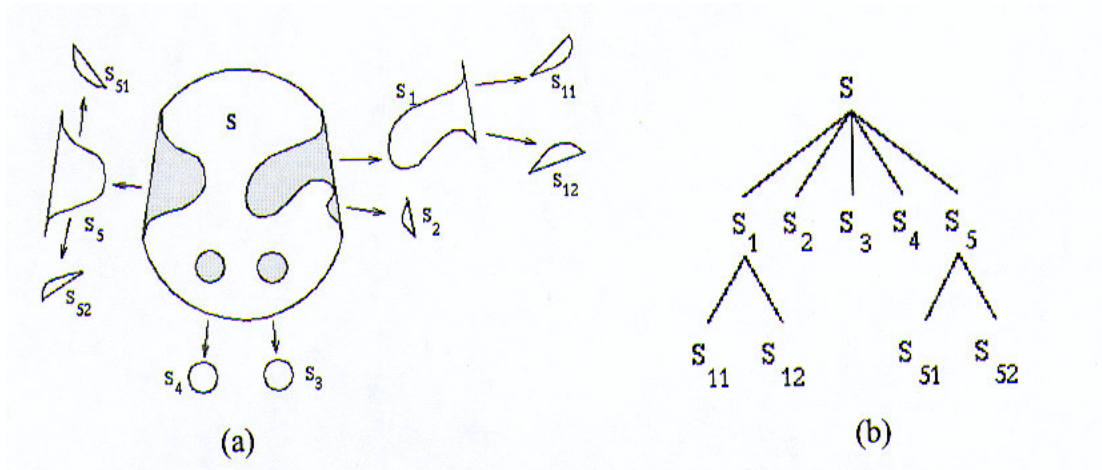


Figure 2.6 (a) L'enveloppe convexe et ses concavités (b) l'arbre de représentation de la concavité de l'enveloppe convexe

2.3.2.2.2 L'axe médian

Comme l'enveloppe convexe, La région squelettique peut également être employée pour la représentation et la description de silhouettes. Un squelette peut être défini comme un ensemble connecté de lignes médianes le long des membres de la silhouette [58].

Par exemple, dans le cas des caractères épais écrits à la main, on peut supposer que le squelette soit le chemin parcouru par le stylo. La base de l'idée du squelette est qu'on élimine l'information superflue, tout en retenant seulement l'information topologique au sujet de la structure de l'objet qui peut aider pour la reconnaissance.

Les méthodes squelettiques sont représentées par la transformée de l'axe médian de Blum (MAT = medial axis transform) [34].

L'axe médian est le lieu des centres des disques maximums qui entrent dans la silhouette comme illustré par la figure 2.7. La ligne en gras sur la figure est le squelette du rectangle de la silhouette. Le squelette peut alors être décomposé en segments et représenté comme un graphe selon certains critères. L'appariement entre les silhouettes devient un problème d'appariement de graphes [150].

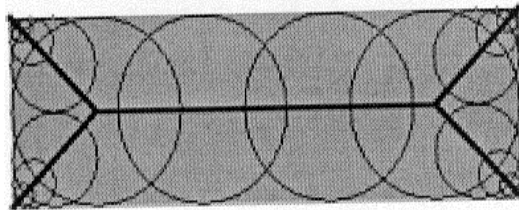


Figure 2.7 Construction de l'axe médian d'une silhouette d'un rectangle [150]

2.3.3 Utilisation de la silhouette pour la reconnaissance d'objets

Nous venons de présenter plusieurs techniques qui utilisent la silhouette pour la représentation des modèles d'objets.

Généralement la représentation du modèle peut être basée sur le contour externe ou sur les régions internes. La représentation du modèle à partir du contour va dans le même sens que la manière dont se fait la perception des objets par l'homme.

L'utilisation de la silhouette dans la reconnaissance d'objets peut être décomposée en :

- L'extraction de la silhouette à partir de l'image,
- La décomposition de la silhouette pour extraire les primitives les plus significatives (segmentation) et
- L'utilisation de ces primitives pour la reconnaissance.

La détection des contours implique la recherche des discontinuités locales de la fonction des niveaux de gris de l'image. La segmentation des contours consiste à faire des approximations des contours par des représentations analytiques, telles que des droites ou des

coniques. L'extraction de régions revient à déterminer des zones homogènes en niveaux de gris de l'image. Par exemple, dans le cas d'images réelles, les contours correspondent aux frontières des objets et les régions à leurs surfaces. Ces deux approches «contour» et «région» sont duales en ce sens qu'une région définit une ligne par son contour, et qu'une ligne fermée définit une région. Elles amènent cependant à des algorithmes complètement différents et ne fournissent pas les mêmes résultats. Cette dualité est cependant peu exploitée dans la plupart des méthodes existantes.

Un autre aspect de la segmentation est celui qui consiste à retrouver la géométrie des objets à partir des images. On obtient ainsi des représentations intrinsèques, aisément manipulables et utilisables, à partir de la réalité physique induite par l'image. Pour obtenir ces caractéristiques géométriques on est souvent conduit à définir une suite hiérarchique de la représentation de l'information image permettant finalement d'obtenir des indices visuels servant à résoudre une tâche donnée.

Le problème de la modélisation de contours, l'extraction et la détection ont souvent été traités séparément dans la vision par ordinateur, vu la complexité de chaque partie bien qu'il existe dans la littérature des modèles qui les traitent ensemble : Le modèle du snake [106] par exemple.

Une fois la silhouette obtenue, la plupart des systèmes basés sur les silhouettes passent par un deuxième niveau de traitement qui consiste à passer de la représentation graphique vers une représentation symbolique. C'est dans ce but que des langages de description de silhouettes sont développés pour permettre une manipulation beaucoup plus souple de l'image. Ces langages se basent sur la silhouette en extrayant toutes les primitives nécessaires pour la mise en correspondance. Ces primitives sont représentées par un ensemble de symboles qu'on désigne par l'alphabet du langage.

2.4 Une approche textuelle pour la représentation des silhouettes

Dans le but de la description des silhouettes, plusieurs travaux de recherche ont été réalisés, et plusieurs langages ont été développés, on peut citer quelques uns :

- Le langage FOHDEL pour décrire les symboles manuscrits [135].
- Le langage proposé par R.DAVIS [58] pour la description de l'aspect de croquis des symboles.
- Le langage LWDOS (Language for Writing Descriptors of Outline Shape) développé au sein de l'équipe Vision du laboratoire L.R.I.A de l'USTHB [120, 36].

Les deux premiers langages traitent uniquement des objets de formes régulières. Le langage LWDOS [120] a été développé afin de pouvoir décrire textuellement des silhouettes d'objets 3D de forme quelconque. L'avantage de cette technique réside dans le fait qu'elle est indépendante de la rotation de l'objet dans l'image.

Des améliorations ont été introduites au langage LWDOS, et ce en intégrant le XML qui est utilisé pour une meilleure représentation des silhouettes [121].

2.4.1 Description des Silhouettes suivant le langage LWDOS

LWDOS [120, 36] est basé sur l'utilisation du rectangle d'aire minimum (RM) englobant le contour de la silhouette [80].

Soit (OXY) le repère associé au rectangle RM choisi tel que l'angle supérieur gauche du rectangle soit l'origine O du repère (OXY). Les axes OX (resp. OY) correspondent à la largeur (resp. à la longueur) de la silhouette [120] (figure 2.8).

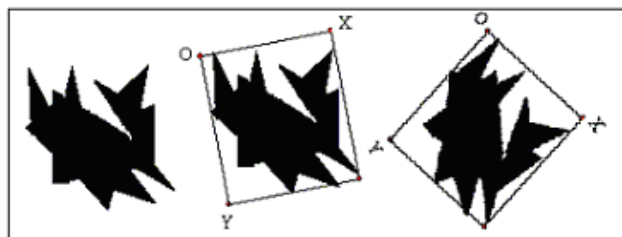


Figure 2.8 La silhouette initiale, le rectangle minimum RM, et la rotation de la silhouette

Avec le choix de ce repère, la forme du contour sera invariante à la rotation.

La première étape de la description consiste à décomposer la silhouette en parties et lignes de séparation (lignes de jonction et lignes de disjonction). Pour ce faire, on effectue le balayage de la silhouette du haut en bas suivant la direction horizontale du rectangle (RM), et on cherche les points de concavité pour lesquels le contour de la silhouette change de direction : haut-bas-haut ou bien bas-haut-bas (figure 2.9).

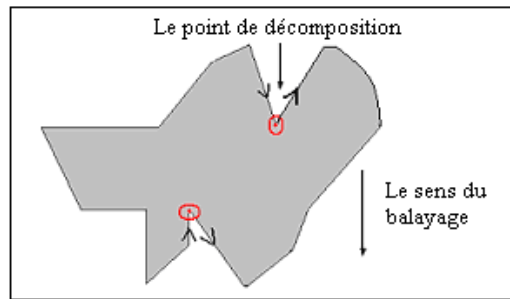


Figure 2.9 Localisation des points de décomposition

Une fois les points de concavité localisés sur la frontière de la silhouette, cette dernière sera décomposée en parties et en lignes de séparation. Soit deux parties (ou plus) sont liées à une troisième par une ligne de jonction, ou bien une partie est liée à deux autres parties (ou plus) par une ligne de disjonction. Ce processus appliqué à la figure 2.9 produit cinq parties (P1 jusqu'à P5), une ligne de jonction (JL1) et une ligne de disjonction (DJL1). Les parties et les lignes sont numérotées de haut en bas et de gauche à droite (figure 2.10)

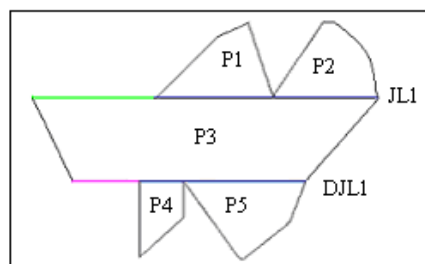


Figure 2.10 Parties et Lignes de séparation

2.4.1.1 Descripteur Global

Le descripteur global d'une silhouette décrit conformément au langage LWDOS [120] est un descripteur dans lequel apparaissent seulement les parties ainsi que les lignes de jonction et

de disjonction sans donner leurs descriptions géométriques. Par exemple, le descripteur global LWDOS de la silhouette de la figure 2.10 est :

$$[P1 \cup P2 \oplus \uparrow JL1 \oplus P3 \oplus \downarrow DJL1 \oplus P4 \cup P5].$$

JL_i indique la $i^{\text{ème}}$ ligne de jonction rencontrée en balayant l'image de haut vers le bas et de gauche à droite. DJL_j indique la $j^{\text{ème}}$ ligne de disjonction. Des silhouettes différentes peuvent évidemment avoir le même descripteur global.

2.4.1.2 Descripteur Détaillé

Afin d'enlever les ambiguïtés dues au descripteur global qui peut décrire des silhouettes différentes, la deuxième étape de la description consiste à la description géométrique détaillée de chaque élément (chaque partie et ligne) dans le but de garantir une description unique caractérisant exclusivement une forme donnée.

La description détaillée est basée sur les caractéristiques géométriques de tous les éléments de la silhouette définies ci-dessous :

- Une partie est décrite par ses frontières gauche et droite.
- Une frontière est une succession de primitives.
- Une primitive peut avoir l'une des formes suivantes: Une courbe convexe (C_v), Une courbe concave (C_c), ou bien un segment de droite (L_n) [120]. Une primitive est caractérisée aussi par son angle d'inclinaison et sa taille.
- La taille d'une primitive correspond à sa hauteur sauf si la primitive est horizontale (voir figure 2.11).
- Lorsque la primitive est une courbe convexe (resp. une courbe concave), elle est caractérisée aussi par son degré de convexité (resp. degré de concavité). (figure 2.12).
- La longueur et la largeur d'une silhouette sont données relativement au rectangle d'aire minimum (RM) qui englobe la forme.

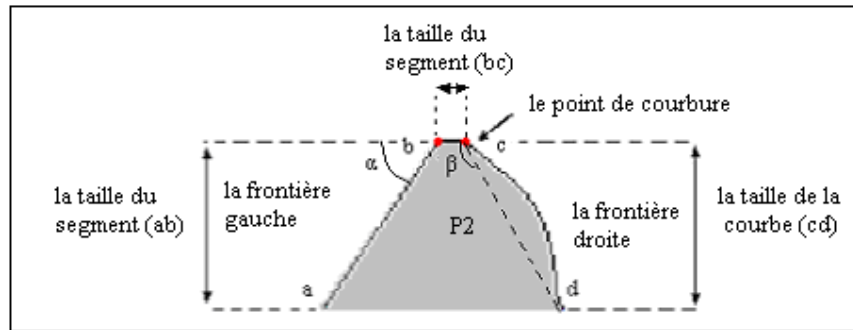


Figure 2.11 Descripteur détaillé d'une partie

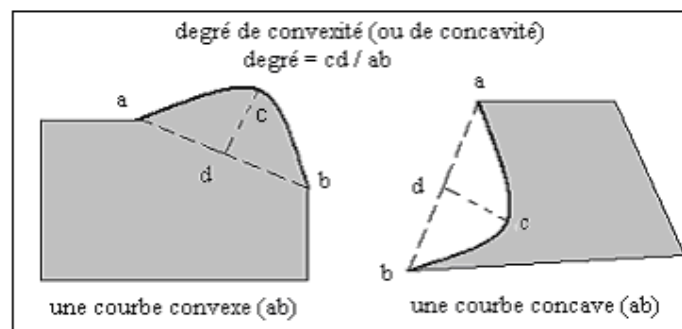


Figure 2.12 Degrés de convexité et de concavité

A titre d'exemple, la frontière gauche de la partie P2 (figure 2.11) est constituée d'un segment de droite avec un angle d'inclinaison $\alpha = 43^\circ$ et une longueur de 109 pixels. La frontière droite est composée d'un segment de droite ayant un angle d'inclinaison de 180° (segment horizontal) et une longueur de 17 pixels, suivi par une courbe convexe (notée Cv) ayant un degré de convexité de 0.09 (9%), un angle d'inclinaison de $\beta = 108^\circ$ et une longueur de 109 pixels.

Le langage LWDOS utilise les dimensions relatives contrairement à XLWDOS (voir plus loin) qui utilise les dimensions absolues. Les dimensions relatives sont exprimées en pourcentage par rapport à la taille du rectangle minimum (RM).

Le contour d'une forme est découpé en contours élémentaires (primitives) après la détermination des points de courbures (points d'inflexion) localisés par l'un des algorithmes les plus performants [50]. Le nombre de ces points est proportionnel à la précision de la description (voir figure 2.13).

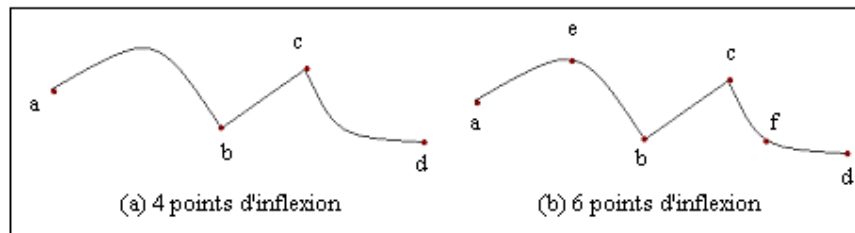


Figure 2.13 les points d'inflexion

Les lignes de jonction et de disjonction sont décomposées en segments élémentaires. Chaque segment est décrit par sa longueur et son type. Il existe trois types de segments: "Junction" (ou bien "shared") si le segment est adjacent à deux parties, "Free-High" si le segment est adjacent à la partie d'en haut (la partie située avant la ligne de jonction), et "Free-Low" si le segment est adjacent à la partie d'en bas (la partie située après la ligne de disjonction) (figure 2.14).

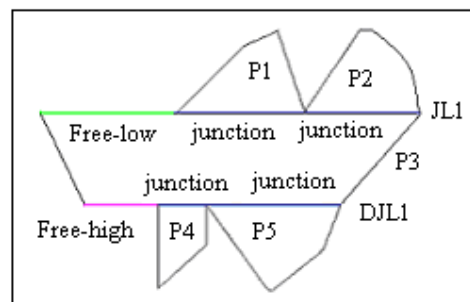


Figure 2.14 Description des lignes de séparation

Par exemple, la ligne de jonction JL1 est décomposé en trois segments: le premier est un segment "free-low" parce qu'il est adjacent à la partie P3. Le second est un segment "junction" car il est commun aux deux parties P1 et P3. Le dernier est aussi un segment "junction" car il est commun aux deux parties P2 et P3.

De même, la ligne de disjonction DJL1 contient trois segments: "free-high", "junction" et "junction".

La taille de chaque segment est exprimée relativement au rectangle d'aire minimum (RM).

2.4.2 La grammaire du langage LWDOS

La grammaire de LWDOS [120] est donnée par le 4-tuple (V_N, V_T, P, S_0) .

Le vocabulaire non terminal V_N du langage est donné par :

$V_N = \{S_0, \langle \text{Composed_part} \rangle, \langle \text{United_part} \rangle, \langle \text{Part} \rangle, \langle \text{Left_Boundary} \rangle, \langle \text{Right_Boundary} \rangle, \langle \text{Contour_descriptor} \rangle, \langle \text{Contour_geometry} \rangle, \langle \text{Relative_length} \rangle, \langle \text{Junction_line} \rangle, \langle \text{Disjunction_line} \rangle, \langle \text{State} \rangle\}$.

Part désigne une partie

- *Junction_Line* (respectivement *Disjunction_Line*) désigne la ligne de jonction (respectivement disjonction)
- *United_Part* désigne un ensemble de parties qui sont liées à une partie à travers une ligne de jonction ou disjonction entre parties
- *Composed_Part* désigne une partie composée de *United_Part*, une ligne de jonction ou disjonction, et une *Part*
- *Left_Boundary* (respectivement *Right_Boundary*) désigne la frontière gauche (respectivement droite)
- *Contour_Descriptor* désigne le descripteur du contour
- *Contour_Geometry* désigne la géométrie du contour
- *State* désigne l'état (type) du segment de jonction ou disjonction
- *Relative_Length* désigne la longueur (dimension) relative.

Le vocabulaire terminal de LWDOS est donné par :

$V_T = \{ \oplus, \otimes, [,], \{, \}, \uparrow, \downarrow, U, \text{Junction}, \text{Free_High}, \text{Free_Low}, \text{very_short}, \text{short},$

$\text{less_short}, \text{middle}, \text{enough_long}, \text{long}, \text{very_long}, \text{entire} \} \cup \text{Set_attr_feat}$ où
 $\text{Set_attr_feat} = \{ \text{Ln_Vrt}, \text{Wcv_Vrt}, \text{Cv_Vrt}, \text{Scv_Vrt}, \text{Wcc_Vrt}, \text{Cc_Vrt}, \text{Scc_Vrt}, \text{Ln_Hor}, \text{Wcv_Hor}, \text{Cv_Hor}, \text{Scv_Hor}, \text{Wcc_Hor}, \text{Cc_Hor}, \text{Scc_Hor}, \text{Ln_Wil}, \text{Wcv_Wil},$

Cv_Wil, Scv_Wil, Wcc_Wil, Cc_Wil, Scc_Wil, Ln_Il, Wcv_Il, Cv_Il, Scv_Il, Wcc_Il, Cc_Il, Scc_Il, Ln_Sil, Wcv_Sil, Cv_Sil, Scv_Sil, Wcc_Sil, Cc_Sil, Scc_Sil, Ln_Wir, Wcv_Wir, Cv_Wir, Scv_Wir, Wcc_Wir, Cc_Wir, Scc_Wir, Ln_Ir, Wcv_Ir, Cv_Ir, Scv_Ir, Wcc_Ir, Cc_Ir, Scc_Ir, Ln_Sir, Wcv_Sir, Cv_Sir, Scv_Sir, Wcc_Sir, Cc_Sir, Scc_Sir

S_0 étant l'axiome de la grammaire, l'ensemble des règles de production de cette dernière est :

$S_0 \rightarrow \langle \text{Composed_part} \rangle$

$\langle \text{Composed_part} \rangle \rightarrow [\langle \text{United_part} \rangle \oplus \uparrow \langle \text{Junction_line} \rangle \oplus \langle \text{Part} \rangle] / [\langle \text{Part} \rangle \oplus \downarrow \langle \text{Disjunction_line} \rangle \oplus \langle \text{United_part} \rangle]$

$\langle \text{United_part} \rangle \rightarrow \langle \text{Part} \rangle \cup \langle \text{United-part} \rangle / \langle \text{Part} \rangle$

$\langle \text{Part} \rangle \rightarrow \{ \langle \text{Left_Boundary} \rangle \otimes \langle \text{Right_Boundary} \rangle \} / \langle \text{Composed_part} \rangle / \langle \text{United_part} \rangle$

$\langle \text{Left_Boundary} \rangle \rightarrow \langle \text{Contour_descriptor} \rangle \langle \text{Left_Boundary} \rangle / \langle \text{contour_descriptor} \rangle$

$\langle \text{Right_Boundary} \rangle \rightarrow \langle \text{Contour_descriptor} \rangle \langle \text{Right_Boundary} \rangle / \langle \text{Contour_descriptor} \rangle$

$\langle \text{Contour_descriptor} \rangle \rightarrow \langle \text{Contour_geometry} \rangle \langle \text{Relative_length} \rangle$

$\langle \text{Contour_geometry} \rangle \rightarrow Ln_Vrt / Wcv_Vrt / \dots / Wcc_Sir / Cc_Sir / Scc_Sir$

$\langle \text{Junction_line} \rangle \rightarrow \langle \text{State} \rangle \langle \text{Relative_length} \rangle /$

$\langle \text{State} \rangle \langle \text{Relative_length} \rangle \langle \text{Junction_line} \rangle$

$\langle \text{Disjunction_line} \rangle \rightarrow \langle \text{State} \rangle \langle \text{Relative_length} \rangle / \langle \text{State} \rangle \langle \text{Relative_length} \rangle$

$\langle \text{Disjunction_line} \rangle$

$\langle \text{State} \rangle \rightarrow \text{Junction} / \text{Free_High} / \text{Free_Low}$

$\langle \text{Relative_length} \rangle \rightarrow \text{very_short} / \text{short} / \text{less_short} / \text{middle} / \text{enough_long} / \text{long}$

$\text{very_long} / \text{entire}$

On peut énumérer les propriétés suivantes des descripteurs écrits avec LWDOS :

- ✓ L'utilisation des longueurs (dimensions) relatives produit des descripteurs invariants au changement d'échelle,
- ✓ L'invariance des descripteurs LWDOS à la rotation est due à l'utilisation du référentiel attaché au rectangle minimum englobant la silhouette,
- ✓ Les symboles délimiteurs {, }, [,] utilisés facilitent l'indexation des descripteurs.

2.4.2.1 Exemple

La silhouette de la figure 2.15 est partitionnée en 7 parties (P1 jusqu'à P7), une ligne de jonction JL1 et une ligne de disjonction DJL1.

Pour avoir le descripteur LWDOS de cette silhouette, nous suivons les étapes ci dessous :

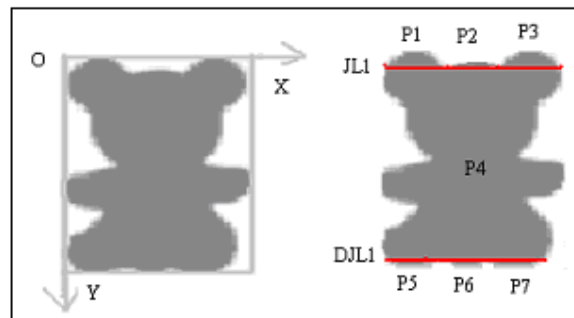


Figure 2.15 Décomposition de la Silhouette selon LWDOS

Etape 1 : Obtenir les composants de la silhouette en la balayant suivant l'axe OX, de haut en bas. Nous aurons trois parties P1, P2, P3, une ligne de jonction JL1 une partie P4, une ligne de disjonction DJL1 et trois autres parties P5, P6, P7.

Etape 2 : Pour chaque partie, extraire les caractéristiques de contours ainsi que leurs attributs.

Etape 3 : Ecrire la description de chaque partie selon la syntaxe du LWDOS.

Les attributs *Very_short*, *short*, *less_short*, *middle*, *enough_long*, *long*, *very_long* et *entire* des dimensions relatives correspondent aux intervalles suivants: [0 0.1], [0.1 0.2], [0.2 0.4], [0.4 0.55], [0.55 0.7], [0.7 0.85], [0.85 0.99] et [1 1]. Les attributs *Weakly convex* or

Weakly concave, *Convex* or *Concave*, *Strongly convex* or *Strongly concave* correspondent respectivement aux conditions: ($r/L \leq 1/4$), ($1/4 < r/L \leq 3/4$) and ($3/4 < r/L$). où r et L correspondent aux longueurs des segments (cd) et (ab) utilisés pour calculer le degré de convexité ou de concavité (voir figure 2.12). Les attributs: *Horizontal*, *Vertical*, *Weakly inclined*, *Inclined* et *Strongly inclined* correspondent aux intervalles suivants pour les angles d'inclinaison: 0° , 90° , $]0^\circ, 30^\circ]$, $]30^\circ, 60^\circ]$, $]60^\circ, 90^\circ[$.

Les descriptions obtenues pour chaque partie sont donnés ci-dessous. (Le symbole “ \otimes ” est un séparateur entre la frontière gauche (left boundary) et la frontière droite (right boundary) d'une partie).

P1 \rightarrow { *Wcv_Ill* very_short \otimes *Wcv_Ir* very_short }

P2 \rightarrow { *Wcv_Wil* very_short \otimes *Wcv_Wir* very_short }

P3 \rightarrow { *Wcv_Ill* very_short \otimes *Wcv_Ir* very_short }

P4 \rightarrow { *Wcv_Ir* short *Wcv_Sir* less_short *Wcv_Ill* very_short *Cv_Ir* short *Ln_Ill* short *Cv_Vrt* short \otimes *Wcv_Ill* short *Wcv_Sil* less_short *Wcv_Ir* very_short *Cv_Ill* short *Ln_Ir* short *Cv_Vrt* short }

P5 \rightarrow { *Wcv_Wir* very_short \otimes *Wcv_Wil* very_short }

P6 \rightarrow { *Wcv_Ir* very_short \otimes *Wcv_Ill* very_short },

P7 \rightarrow { *Wcv_Wir* very_short \otimes *Wcv_Wil* very_short }

Etape 4 : Ecrire la description des lignes de jonction et de disjonction selon la syntaxe du LWDOS.

JL1 \rightarrow *junction less_short*, *junction less_short*, *junction less_short*

DJL1 \rightarrow *junction less_short*, *junction less_short*, *junction less_short*

Etape 5 : Ecrire le descripteur global en utilisant la syntaxe du LWDOS.

Silhouette $\rightarrow [[P1 \cup P2 \cup P3 \oplus \uparrow JL1 \oplus P4] \oplus \downarrow DJL1 \oplus P5 \cup P6 \cup P7]$.

Etape 6 : Obtenir le descripteur complet en remplaçant toutes les parties et lignes (de jonction et de disjonction) par leurs syntaxes correspondantes. Cela nous donne :

$[[\{ Wcv_Il \text{ very_short} \otimes Wcv_Ir \text{ very_short} \} \cup \{ Wcv_Wil \text{ very_short} \otimes Wcv_Wir \text{ very_short} \} \cup \{ Wcv_Il \text{ very_short} \otimes Wcv_Ir \text{ very_short} \} \oplus \uparrow \textit{junction less_short, junction less_short, junction less_short} \oplus \{ Wcv_Ir \text{ short } Wcv_Sir \text{ less_short } Wcv_Il \text{ very_short } Cv_Ir \text{ short } Ln_Il \text{ short } Cv_Vrt \text{ short} \} \otimes Wcv_Il \text{ short } Wcv_Sil \text{ less_short } Wcv_Ir \text{ very_short } Cv_Il \text{ short } Ln_Ir \text{ short } Cv_Vrt \text{ short} \}] \oplus \downarrow \textit{junction less_short, junction less_short, junction less_short} \oplus \{ Wcv_Wir \text{ very_short} \otimes Wcv_Wil \text{ very_short} \} \cup \{ Wcv_Ir \text{ very_short} \otimes Wcv_Il \text{ very_short} \} \cup \{ Wcv_Wir \text{ very_short} \otimes Wcv_Wil \text{ very_short} \}]$.

2.4.3 Le langage XLWDOS

XML (EXtensible Markup Language) traduit en Langage à balises étendu, ou Langage à balises extensible) est en quelque sorte un langage HTML amélioré permettant de définir de nouvelles balises. Il s'agit effectivement d'un langage permettant de mettre en forme des documents grâce à des balises (markup) [121].

Contrairement à HTML, qui est considéré comme un langage défini et figé (avec un nombre de balises limité), XML peut être considéré comme un métalangage permettant de définir d'autres langages, c'est-à-dire définir de nouvelles balises permettant de décrire la présentation d'un texte.

La force de XML réside dans sa capacité à pouvoir décrire n'importe quel domaine de données grâce à son extensibilité. Il permet de structurer, poser le vocabulaire et la syntaxe des données qu'il va contenir.

En réalité les balises XML décrivent le contenu plutôt que la représentation (contrairement à HTML). Ainsi, XML permet de séparer le contenu de la représentation, ce qui permet par exemple d'afficher un même document sur des applications ou des périphériques différents sans pour autant nécessiter de créer autant de versions du document que l'en nécessitent les représentations.

En utilisant toujours le langage LWDOS, mais en y introduisant le XML, on aura une nouvelle grammaire, donc une nouvelle écriture. En plus des différences de forme entre LWDOS et XLWDOS, le premier langage consiste à utiliser les descriptions relatives des formes, tandis que le deuxième utilise les descriptions absolues, en donnant les valeurs exactes pour les primitives des images, comme les longueurs, les angles etc...

Les longueurs relatives sont fortement conseillées pour les applications de la vision par ordinateur, où on effectue une description et comparaison « grossières » des formes, notamment pour le suivi d'objets en mouvement et aussi pour l'indexation d'images tandis que les dimensions absolues seront utilisées pour les applications multimédia, ou en reconnaissance de forme (dans la vision artificielle) lorsqu'il s'agit de trouver des mesures de similarité entre les images.

Voici la grammaire écrite en utilisant XLWDOS pour chaque composant d'une silhouette :

a) Ecriture XML d'une partie

Part → <P PartNumber> Left-Boundary Right-Boundary </P PartNumber>

Left-Boundary → <L> Contour 1 Contour n </L>

Right-Boundary → <R> Contour 1 Contour m </R>

Contour i → Cv Convexity-degree Inclination-angle Length /
Cc Concavity-degree Inclination-angle Length /
r Inclination-angle Length

Avec :

- PartNumber : le numéro de la partie décrite.
- <P PartNumber> et </P PartNumber> : Les marques de début et de fin de la partie ayant comme numéro PartNumber.
- <L>, </L> et <R>, </R>: Les marques de début et de fin des frontières gauche et droite respectivement.

- Cv, Cc, r : indiquent respectivement convexe, concave et droite.
- Left-Boundary et Right-Boundary : Correspond au contour gauche et droit d'une partie.
- Contour i : Correspond aux caractéristiques d'un contour élémentaire.

b) Ecriture XML d'une ligne de jonction et de disjonction

Une ligne de jonction ou de disjonction est décrite comme suit :

Junction line → <J JunctionNumber > Segment 1... Segmentn </J JunctionNumber >

Disjunction line → <D DisjunctionNumber > Segment 1 Segmentm
</D DisjunctionNumber >

Segment i → s High-Part-Number Low-Part-Number Length /

w Low-Part-Number Length /

h High-Part-Number Length

Avec :

- <J JunctionNumber >, </J JunctionNumber >: Les marques de début et de fin des ligne de jonction ayant comme numéro JunctionNumber.
- <D DisjunctionNumber >, </D DisjunctionNumber >: Les marques de début et de fin de ligne de disjonction ayant comme numéro « Disjunction Number ».
- Segment i : Correspond aux caractéristiques du segment qui compose la ligne de jonction ou de disjonction.
- s, w, h : indiquent respectivement Shared, Free-Low et Free-High.

c) Ecriture XML d'une partie composée

Une partie composée est décrite à l'aide du non terminal ComposedPart comme suit :

Composed part → <CP> P 1, P 2 P n JL P n+1 </CP> /
 <CP> P 1 DJL P 2, P 3 P n </CP>

Avec :

- <CP>, </CP>: Les marques de début et de fin de la partie composée.
- P i : Correspond à la description de la partie numéro i.
- JL et DJL : Indique le type de ligne de partie (Jonction ou Disjonction).

Pour la description des silhouettes, on utilise la syntaxe suivante :

Silhouette → <DXLWDOS> <Name> <nom silhouette> </Name> <Length> <taille>
 </Length> <partie composée> </DXLWDOS>

2.4.3.1 Exemple

Le descripteur global de la silhouette de la figure 2.10 en utilisant la syntaxe XLWDOS est: <CP><CP>P₁ P₂ J₁ P₃ </CP> D₁ P₄ P₅</CP>

<P1> <L>r 32 29 r43 58 </L> <R>r109 79 </R></P1>
 <P2> <L> r 43 109 </L> <R>r108 17 CV 9 108 109 </R></P2>
 <P3> <L>r116 80 </L> <R>r49 82 </R></P3>
 <P4> <L>r90 70 </L> <R>r90 35r42 36 </R></P4>
 <P5> <L>r126 76 </L> <R>r69 41r39 32 </R></P5>
 <J1> W P3 108 S P1 P3 112 S P2 P3 98 </J1>
 <D1> H P3 66 S P3 P4 40 S P3 P5 115 </D1>

Pour obtenir la description détaillée, nous remplaçons, dans le descripteur global, chaque partie et chaque ligne de jonction ou de disjonction par sa description. La description détaillée de la silhouette devient donc:

<CP>
 <CP> <P1> <L>r 32 29 r43 58 </L> <R>r109 79 </R></P1>
 <P2> <L> r 43 109 </L> <R>r108 17 CV 9 108 79 </R></P2>

```
<J1> W P3 108 S P1 P3 112 S P2 P3 98 </J1>
<P3> <L>r116 80 </L> <R>r49 82 </R></P3>
</CP>
<D1>    H P3 66 S P3 P4 40 S P3 P5 115 </D1>
<P4> <L>r90 70 </L> <R>r90 35r42 36 </R></P4>
<P5> <L>r126 76 </L> <R>r69 41r39 32 </R></P5>
</CP>
```

2.5 Conclusion

Des techniques existantes sur la représentation de la silhouette ont été passées en revue dans ce chapitre. Il y'a deux types d'approches : une basée sur le contour et une basée sur la région. Les approches basées sur le contour sont plus populaires que les approches basées sur la région, parce que l'on croit que l'être humain distingue les formes principalement par leurs caractéristiques de contour. Les méthodes basées sur le contour impliquent habituellement moins de calcul que les méthodes basées sur la région. Les méthodes basées sur la région sont généralement plus robustes, cependant, elles impliquent plus de calcul, et les descripteurs de région de silhouettes ont besoin de plus d'espace de stockage que les descripteurs de contours.

Nous avons présenté et illustré avec des exemples les langages LWDOS et XLWDOS, qui permettent de générer les descripteurs textuels décrivant des silhouettes d'objets 3D. Ces descriptions apportent une grande flexibilité pour l'extraction des primitives nécessaires pour minimiser l'espace mémoire lors du stockage des descripteurs au lieu des images correspondantes.

Nous verrons par la suite que les descripteurs obtenus peuvent changer dans leur structure à cause du bruit ou à cause d'un léger changement dans la prise de vue des images, ce qui rendrait une comparaison « brute » des descripteurs une opération difficile (voire impossible).

Dans le prochain chapitre, nous prendrons en considération l'influence du bruit sur la description des silhouettes, pour mettre en correspondance deux descripteurs textuels.

Chapitre 3

Mise en correspondance des descripteurs textuels de silhouettes

3.1 Introduction

Plusieurs travaux ont été proposés pour la mise en correspondance de descripteurs LWDOS de formes [36, 37] opérant directement sur les descripteurs et ne tenant pas compte des déformations des frontières de la forme requête.

Nous montrons dans le paragraphe suivant qu'une déformation du contour de la forme requête peut engendrer une modification de son descripteur, alors que celle-ci est similaire à la forme modèle. Nous proposons ensuite une méthode de mise en correspondance de descripteurs LWDOS de silhouettes d'objets en tenant compte des déformations pouvant altérer les frontières.

Cette mise en correspondance concernera d'abord la structure globale des descripteurs. La seconde étape concernera les détails des éléments des descripteurs (géométrie des parties et leur positionnement par rapport aux lignes de séparation).

3.2 Effet de la déformation du contour externe d'une forme sur son descripteur LWDOS

Une déformation du contour externe d'une forme peut engendrer un changement du descripteur. Une comparaison de celui-ci avec le descripteur modèle nous ramènerait à conclure sévèrement que les objets représentés sont différents. Dans le but de résoudre ce problème et en tenir compte dans la comparaison de descripteurs, nous avons proposé une méthode permettant de transformer les descripteurs à comparer [4]. En effet si deux descripteurs de deux silhouettes peuvent être transformés en un autre descripteur commun, nous dirons que ces silhouettes représentent le même objet à une tolérance près, c'est donc une comparaison à plusieurs échelles c'est-à-dire à plusieurs niveaux de tolérance.

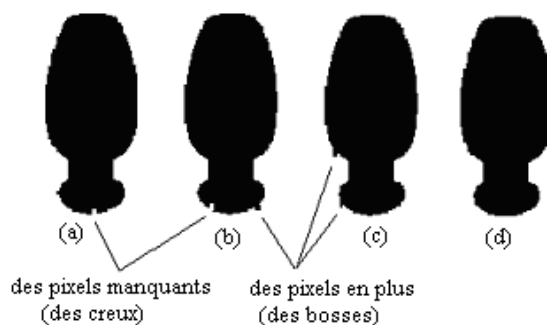


Figure 3.1 Exemple de silhouettes bruitées

Considérons la figure 3.1. Les quatre formes représentent évidemment le même objet malgré les petites différences sur leurs contours. Ces différences correspondent à des déformations de la frontière engendrant de nouvelles concavités (creux) ou convexités (bosses). Les descripteurs LWDOS des quatre formes sont différents. Ce qui complique aussi la représentation c'est que non pas seulement les descripteurs détaillés (qui décrivent la géométrie de la forme) qui sont différents mais aussi la description globale concernant le nombre de parties et de lignes.

En effet les descripteurs globaux des formes (a), (b), (c) et (d) de la figure 3.1 sont respectivement (voir figure 3.2):

- $[P1 \ominus \downarrow DJL1 \oplus P2 \cup P3]$
- $[P1 \ominus \downarrow DJL1 \oplus [P2 \ominus \downarrow DJL2 \oplus P3 \cup P4]] \cup [P5 \ominus \downarrow DJL3 \oplus P6 \cup P7]$
- $[P1 \ominus \downarrow DJL1 \oplus P2 \cup [P3 \ominus \downarrow DJL2 \oplus P4 \cup P5]]$
- $[P1]$

En effet, nous remarquons dans la figure 3.2 un partitionnement différent des silhouettes de la figure 3.1 à cause du bruit.

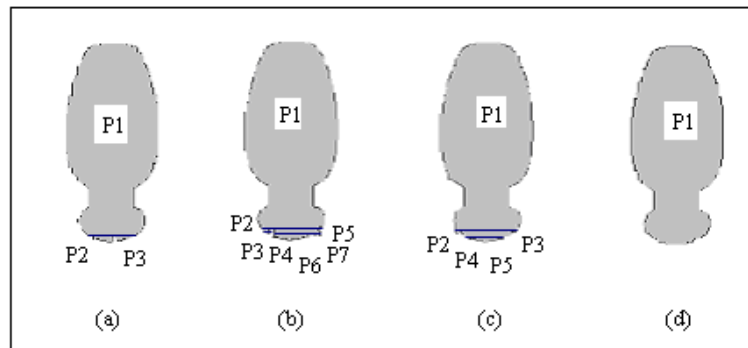


Figure 3.2 Résultats de la décomposition des silhouettes (a), (b), (c) et (d) en parties

En supposant que cette déformation sur le contour externe d'une forme correspond à un bruit, il est possible de remédier à ce problème de plusieurs façons :

La première technique consiste à lisser les frontières permettant ainsi d'atténuer le bruit les entachant [17, 95].

La figure 3.3 montre un exemple de lissage par un filtre gaussien suivi d'un seuillage. Le lissage avec $\sigma=2$ des contours de la pomme et de la voiture a permis de réduire beaucoup de détails sur leurs contours.

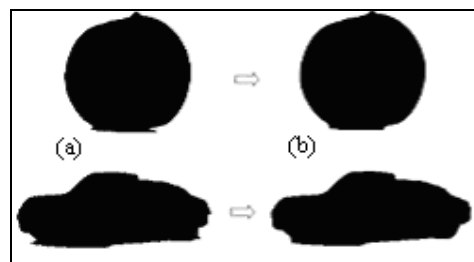


Figure 3.3 Lissage moyennant le filtre Gaussien

La seconde technique consiste à appliquer des transformations au descripteur textuel bruité dans le but d'avoir la même structure globale. Dans notre méthode, toute modification est effectuée sur le code textuel des silhouettes et en aucun cas sur les silhouettes des objets correspondants. La faisabilité de cette approche dépend du degré de la déformation des frontières des silhouettes. Les opérations de transformation opérées doivent concerner uniquement les éléments du descripteur engendrés par un bruit.

3.3 Mise en correspondance des parties composées de descripteurs textuels

3.3.1 Objectif

Il s'agit, disposant de deux descripteurs LWDOS correspondant à deux formes (requête et modèle), de mettre en correspondance leurs parties composées. Sachant que le descripteur lui-même est une partie composée regroupant différentes parties composées, l'objectif de cette étape est de réaliser un appariement structurel des deux descripteurs.

Une fois les parties composées des deux descripteurs appariées à différents niveaux d'imbrication, l'appariement sémantique entre les éléments des parties composées est donc possible commençant par le niveau élémentaire (parties composées dont les éléments sont simples) [5, 7].

Avant d'expliquer la méthode de regroupement et de transformation de descripteurs, nous commencerons par quelques définitions.

Définitions

Nous appelons :

- **Partie principale**, toute partie qui apparaît à droite d'une ligne de jonction ou à gauche d'une ligne de disjonction.
- **Partie secondaire**, toute partie qui apparaît à gauche d'une ligne de jonction ou à droite d'une ligne de disjonction.

- **Longueur des parties secondaires** (notée **LPS**), la longueur maximale des parties secondaires.
- **Ligne de séparation principale** la ligne dont toutes les parties secondaires attachées à celle-ci sont de longueurs supérieures à un seuil LPS_{min} .
- **Ligne de séparation secondaire** la ligne dont toutes les parties secondaires attachées à celle-ci sont de longueurs inférieures à un seuil LPS_{min} .

La figure 3.4 illustre deux exemples de décomposition de silhouettes dont les descripteurs globaux sont:

- $[P1 \cup P2 \cup P3 \oplus \uparrow JL \oplus P4]$ pour la silhouette (a) où P1, P2, P3 sont des parties secondaires alors que P4 est une partie principale.
- $[P1 \oplus \downarrow DJL \oplus P2 \cup P3 \cup P4]$ pour la silhouette (b) où P2, P3, P4 sont des parties secondaires alors que P1 est la partie principale.

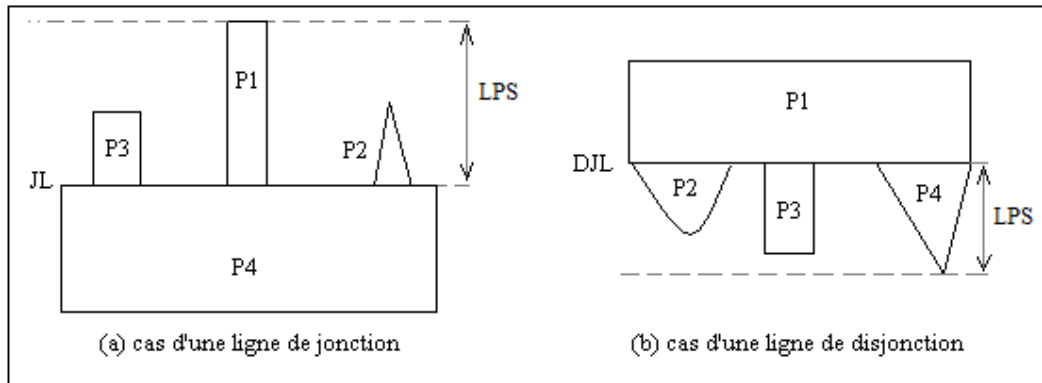


Figure 3.4 Partie principale et parties secondaires

3.3.2 Mise en correspondance des lignes de séparation principales

Une condition nécessaire pour que les descripteurs de deux formes requête et modèle soient en correspondance est que leurs lignes de séparation principales soient en correspondance. Il s'agit donc d'explorer simultanément les descripteurs des deux silhouettes

(requête et modèle) et déterminer les couples de lignes de séparation principales en correspondance.

Pour ce faire, nous utiliserons le résultat suivant :

Proposition

Soient L_i^Q la ligne de séparation numéro i du descripteur requête D^Q et L_j^M la ligne de séparation numéro j du descripteur modèle D^M .

Une condition nécessaire pour que L_i^Q, L_j^M soient en correspondance est que :

- 1) L_i^Q, L_j^M ont la même description textuelle.
- 2) Aucune des parties secondaires attachées à L_i^Q et L_j^M n'est issue d'un bruit. Il suffit pour cela de vérifier que leurs longueurs sont inférieures au seuil LPS_{min} .

La sélection de toutes les paires (L_i^Q, L_j^M) vérifiant les conditions citées ci-dessus, permettra de déterminer les associations entre les parties composées des deux descripteurs.

La procédure **Regroupement_lignes** reçoit en entrée les deux descripteurs textuels D^Q et D^M , et produit en sortie une table de correspondance (Association) entre les lignes de séparation. Elle met aussi le booléen Match à Vrai si pour chaque ligne de séparation principale d'un descripteur est associée une ligne de séparation principale sur l'autre descripteur.

Procédure Regroupement_lignes ($D^Q, D^M, Association, Match$)

Debut

$i=1, j=1$

{N est le nombre de lignes de séparation du descripteur requête}

{M est le nombre de lignes de séparation du descripteur modèle }

Tant que ($i \leq N$) et ($j \leq M$)

faire

Extraire L_i^Q de D^Q

Extraire L_j^M de D^M

```

Apparier_lignes ( $L^Q_i, L^M_j$ , Resultat, bruit)
Si Resultat alors Association ( $i,j$ )=1 ,  $i=i+1, j=j+1$ 
  sinon Association ( $i,j$ )= 0
    Cas bruit égal à
      3 :  $i=i+1, j=j+1$ 
      2 :  $j=j+1$ 
      1 :  $i=i+1$ 
    fincas ;
  Finsi;
Fait;
Si ( $i>N$ ) alors
  Tant que ( $j\leq m$ ) et non ligne_bruitée( $L^M_j, LSP_{min}$ )
    faire  $j=j+1$ 
    fait ;
  Si ( $j>m$ ) alors Match=vrai sinon Match=Faux finsi ;
Finsi ;
Si ( $j>m$ ) alors
  Tant que ( $i\leq n$ ) et non ligne_bruitée ( $L^Q_i, LSP_{min}$ )
    faire  $i=i+1$ 
    fait ;
  Si ( $i>n$ ) alors Match=Vrai sinon Match=Faux finsi;
Finsi ;
Fin.

```

Apparier_lignes (L_i, L_j , Résultat, bruit) est une fonction qui retourne :

- la valeur Vrai pour la variable **Résultat** si les lignes L_i et L_j ont la même description, la valeur Faux dans le cas contraire.
- La valeur **bruit** est égale à :
 - **1** si la ligne L_i est jointe à des parties secondaires dont les longueurs sont inférieures à LSP_{min} .
 - **2** si la ligne L_j est jointe à des parties secondaires dont les longueurs sont inférieures à LSP_{min} .

- **3** si les lignes L_i et L_j sont jointes à des parties secondaires dont les longueurs sont inférieures à LSP_{min} .

Ligne_bruitée (L_i , LSP_{min}) est une fonction qui retourne la valeur vrai si la ligne L_i est jointe à des parties secondaires dont les longueurs sont inférieures à LSP_{min} , et la valeur Faux dans le cas contraire.

S'il reste une ligne de séparation principale sur l'une des silhouettes qui ne possède pas une ligne correspondante sur l'autre silhouette alors cela signifie que les descripteurs à comparer représentent des objets différents.

La figure 3.5 montre un exemple de lignes appariées à la suite de l'application de la procédure **Regroupement_lignes**. Les lignes JL1 et JL'2 seront appariées, de même que pour les lignes DJL1 et DJL'1. Par les lignes JL'1, DJL2 et DJL'2 n'ont pas des lignes correspondantes car elles sont des lignes de séparation secondaires dues à un bruit.

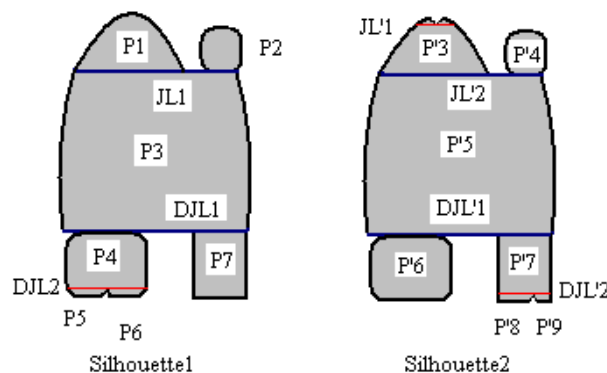


Figure 3.5 Principe de la mise en correspondance des lignes

3.3.3 Mise en relation des parties composées

Une fois les lignes de séparation principales appariées, les parties composées en relation sont mises en couple pour une comparaison sémantique de leurs éléments.

Cette mise en correspondance se fera en commençant par les lignes de séparation principales de niveau zéro (ayant comme éléments des parties simples).

L'algorithme consiste à chercher parmi les associations L_i^Q, L_j^M celles qui sont en relation avec des parties simples. Concrètement, les parties composées correspondantes sont engobées par les symboles "["et "]" les plus internes dans chacun des descripteurs.

Ces parties composées sont ensuite substituées par une partie simple et le processus est réitéré jusqu'à ce que toutes les lignes de séparation aient été traitées. Ainsi en sortie de l'algorithme, des couples de parties composées sont constitués.

La seconde étape, consiste à mettre en correspondance les éléments de chaque paire des parties composées. Le nombre de parties qui sont aux côtés gauches des lignes de séparation doit être identique pour les deux parties composées, de même pour celles qui sont à leur droite. A l'issue de cette étape, une partie simple de la forme requête peut être mise en correspondance avec une partie simple ou composée de la forme modèle et inversement.

A titre d'exemple, donnons le résultat de l'application de ce processus sur les formes illustrées par la figure 3.5.

Le descripteur de la forme requête (silhouette 1) est donné par l'expression:

$[[P1 \cup P2 \oplus \uparrow JL1 \oplus P3] \oplus \downarrow DJL1 \oplus [P4 \oplus \downarrow DJL2 \oplus P5 \cup P6] \cup P7]$

Alors que le descripteur de la silhouette modèle (silhouette 2) de la même figure 3.5 est donné par:

$[[[P'1 \cup P'2 \oplus \uparrow JL'1 \oplus P'3] \cup P'4 \oplus \uparrow JL'2 \oplus P'5] \oplus \downarrow DJL'1 \oplus P'6 \cup [P'7 \oplus \downarrow DJL'2 \oplus P'8 \cup P'9]]$.

Il est évident que ces deux descripteurs sont différents. Le premier contient sept parties et le deuxième en contient neuf, ainsi que les différences quant aux nombres et types de lignes.

A la sortie de l'algorithme de mise en correspondance des lignes de séparation principales, les couples suivants sont obtenus:

- (JL1, JL'2) ayant les descriptions (*junction long, free_low short, junction short*).

- (DJL1 et DJL'1) ayant les descriptions (*junction middle, free_height less short, junction less short*).

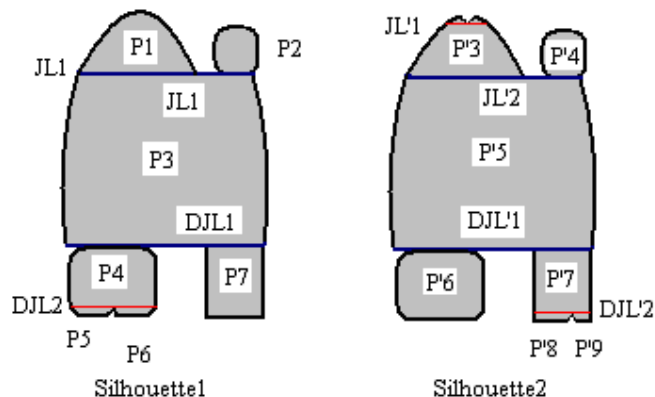


Figure 3.6 Appariement des lignes de séparation

Par contre les lignes de séparation **secondaires** JL'1 et DJL'2 de la silhouette de droite ainsi que la ligne DJL2 de la silhouette de gauche n'ont pas de lignes qui leur correspondent.

En conséquence, les couples suivants de parties composées sont formés (figure 3.7):

$$\begin{aligned}
 & [P1 \cup P2 \oplus \uparrow JL1 \oplus P3], [[P'1 \cup P'2 \oplus \uparrow JL'1 \oplus P'3] \cup P'4 \oplus \uparrow JL'2 \oplus P'5] \\
 & [[P1 \cup P2 \oplus \uparrow JL1 \oplus P3] \oplus \downarrow DJL1 \oplus [P4 \oplus \downarrow DJL2 \oplus P5 \cup P6] \cup P7], \\
 & [[[P'1 \cup P'2 \oplus \uparrow JL'1 \oplus P'3] \cup P'4 \oplus \uparrow JL'2 \oplus P'5] \oplus \downarrow DJL'1 \oplus P'6 \cup [P'7 \oplus \downarrow DJL'2 \oplus P'8 \cup P'9]]
 \end{aligned}$$

Les associations suivantes sont alors dérivées :

$$P1, [P'1 \cup P'2 \oplus \uparrow JL'1 \oplus P'3]$$

$$P2, P'4$$

$$P3, P'5$$

$$[P4 \oplus \downarrow DJL2 \oplus P5 \cup P6], P'6$$

$$P7, [P'7 \oplus \downarrow DJL'2 \oplus P'8 \cup P'9]$$

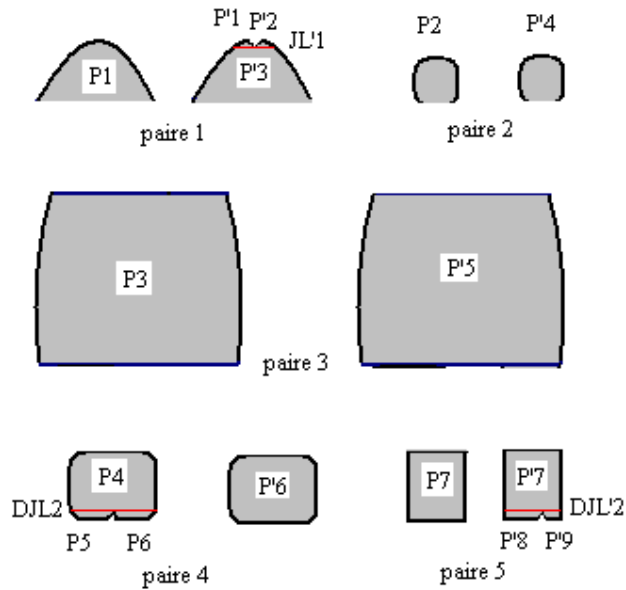


Figure 3.7 Décomposition en paires de parties

3.3.4 Réduction des parties composées associées aux lignes de séparation secondaires

Dans le cas où chacune des parties simples d'un descripteur est associée à une partie simple de l'autre descripteur, le processus de comparaison des deux formes se poursuivra. Dans le cas contraire, une partie simple est associée à une partie composée, le processus de comparaison doit être précédé par l'application des transformations dans le but de réduire ces parties composées en parties simples.

Nous distinguons deux cas de figures :

- Cas où les segments de type « junction » de la ligne de séparation sont courts comparés aux segments de type « free heigh » ou « free low » de la même ligne. Dans ce cas, la partie composée est substituée par une partie simple dont la frontière contiendra la ligne de séparation.

La règle (pour la disjonction) s'écrira alors : $[P_i \oplus \downarrow DJL \oplus P_{j_1} \cup P_{j_2} \dots \cup P_{j_m}] \rightarrow [P'_i]$

- Cas où les segments de type « junction » des lignes de séparation sont longs comparés à ceux de type « free heigh » ou « free low ». Dans ce cas, la partie composée est substituée par une partie simple dont la frontière contiendra la ligne de séparation.

La règle (pour la jonction) s'écrira alors : $[P_{i1} \cup P_{i2} \dots \cup P_{im} \oplus \uparrow JL \oplus P_j] \rightarrow [P'_j]$

Dans les deux cas, les parties secondaires seront donc éliminées du descripteur. Ce traitement est effectué sur la description textuelle, et non sur la forme elle-même. Les figures 3.8 et 3.9 illustrent l'équivalent de ces transformations sur deux exemples de silhouettes.

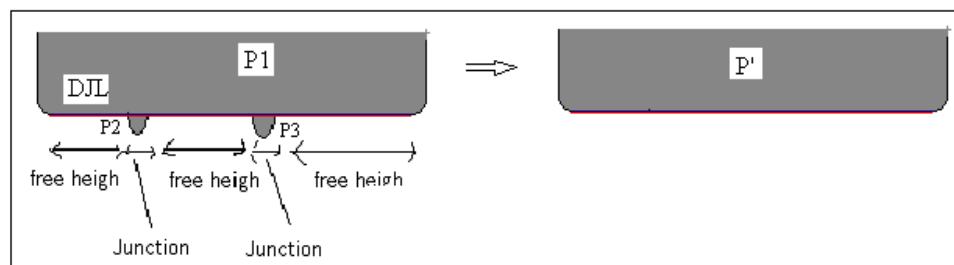


Figure 3.8 Exemple de partie composée et l'équivalent de sa transformation

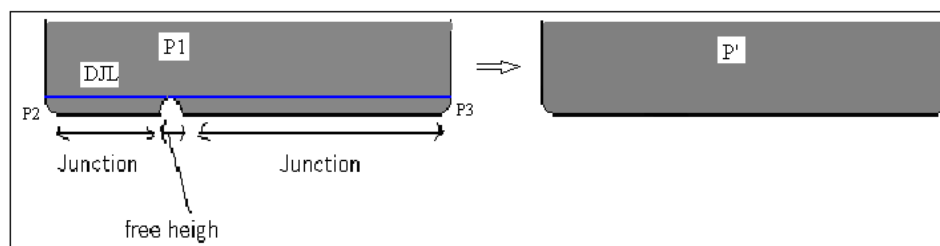


Figure 3.9 Second exemple de partie composée et l'équivalent de sa transformation

Il est important de dire que les descripteurs globaux des deux silhouettes des figures 3.8 et 3.9 sont semblables: $[P1 \oplus \downarrow DL1 \oplus P2 \cup P3]$. Pour cette raison si ce ne sont les détails de la ligne qui nous montrent la différence on serait incapable de parler de parties bruitées ou de creux sur les contours des silhouettes ni de pouvoir distinguer la technique à appliquer pour la résolution d'un tel problème, car rappelons le, nous ne disposons que de descripteurs textuels des objets. Les descripteurs globaux seront donc transformés à une seule partie (P') à la suite du processus de transformation qui aura pour conséquence l'élimination de la ligne de

disjonction à partir du descripteur initial : $[P1 \oplus \downarrow DJL \oplus P2 \cup P3] \rightarrow [P']$ (voir les figures 3.8 et 3.9).

Les tailles des parties bruitées dans la figure 3.8 et celle du vide entre les parties secondaires dans la figure 3.9 sont exagérées, c'est dû au besoin d'expliquer les différents types de segments sur la ligne de liaison. En pratique les tailles des parties à considérer bruitées ne peuvent dépasser quelques pixels comme s'est montrées par la figure 3.10.

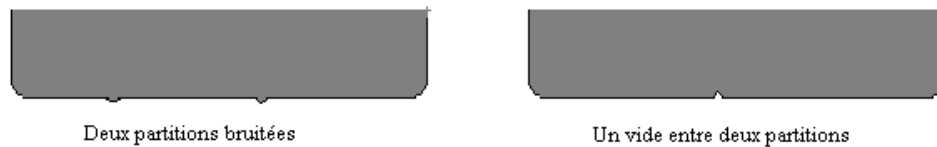


Figure 3.10 Exemples de parties bruitées sur le contour d'une silhouette

Le processus de transformation de descripteurs est récursif, Il est appliqué tant que les parties C_i et C'_i des paires constituées contiennent toujours des sous descripteurs qui vérifient les conditions de réduction. C'est-à-dire tant que les tailles des parties à éliminer restent inférieures au seuil de transformation appelé **seuil_transf**. Ce seuil devrait être choisi en fonction de la hauteur du rectangle minimum englobant la silhouette (par exemple 1/50 de la hauteur du rectangle). Dans notre méthode nous avons pris quasiment la même taille pour les silhouettes sur le repère image quelque soit la nature de l'objet décrit, et nous avons constaté que la valeur maximale *raisonnable* pour le seuil est de l'ordre de 5 pixels.

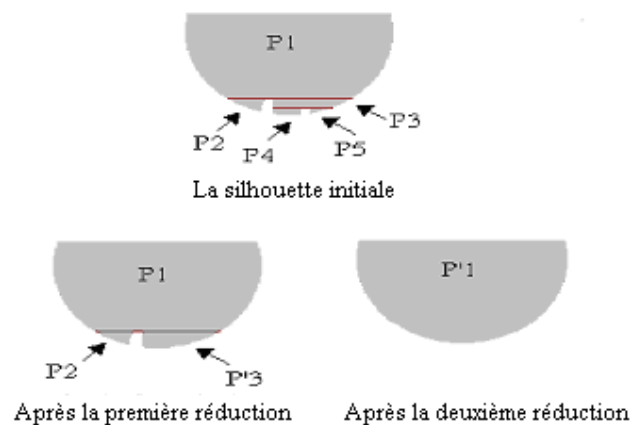


Figure 3.11 Aspect visuel de la réduction de descripteurs

La figure 3.11 montre un exemple d'application récursif du processus de transformation en deux étapes. Pour chacune des étapes, nous constatons que les segments de type « junction » sont supérieurs à ceux de type « free high » ou « free low ».

- Pour la première étape, le prolongement des contours des parties P4 et P5 éliminera le vide entre les parties P4 et P5. La taille du creux est 1 pixel sur la silhouette. La transformation textuelle est donnée par :

$$[P1 \oplus \downarrow DL1 \oplus P2 \cup [P3 \oplus \downarrow DL1 \oplus P4 \cup P5]] \rightarrow [P1 \oplus \downarrow DL1 \oplus P2 \cup P'3].$$

- La deuxième étape procédera sur le résultat de l'étape précédente et éliminera (à partir du descripteur obtenu de la première étape) le vide entre les parties P2 et P'3. La taille du creux est de 3 pixels. La transformation textuelle est donnée par :

$$[P1 \oplus \downarrow DL1 \oplus P2 \cup P'3] \rightarrow [P'1].$$

3.4 Mise en correspondance des parties de descripteurs textuels

3.4.1 Principe de base

Nous disposons de deux descripteurs à comparer dont les structures sont identiques (parties composées appariées). Il s'agit de comparer les descriptions de leurs différentes parties.

Les descriptions des contours de deux parties en correspondance peuvent être différentes en raison d'une légère différence dans leurs contours élémentaires. Une telle déformation ne change pas forcément le descripteur global, cependant elle modifie la géométrie et les dimensions relatives (voir figure 3.12).

Supposons que la silhouette (b) de la figure 3.12 correspond à l'objet modèle et la silhouette (a) de la même figure est la requête. Les descripteurs des deux silhouettes sont différents au niveau des descriptions des frontières droites. En effet, il y a une différence en raison d'un contour élémentaire courbe dans la silhouette (a) qui n'apparaît pas sur la silhouette (b).

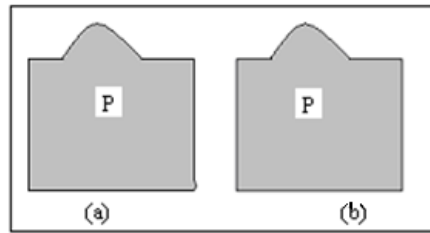


Figure 3.12 Déformation du descripteur détaillé

Lorsque ces différences sont relativement négligeables, nous proposons dans ce qui suit des règles de transformation des expressions du descripteur dans le but d'obtenir un descripteur requête identique au descripteur modèle [6,12]. Ces transformations s'effectueront uniquement sur les descriptions textuelles et en aucun cas sur les silhouettes correspondantes.

3.4.2 Règles de transformation

Notre objectif est de définir un ensemble de règles qui seront appliquées au descripteur requête dont le but d'obtenir la même expression que celui du descripteur modèle [6, 12].

Cet ensemble de règles tient compte de toutes les configurations possibles (types de primitives, orientations, et les dimensions relatives selon la description LWDOS [120]).

Les déformations possibles affectant les descripteurs détaillés sont dues à:

1) Une déformation d'une courbe : Une courbe déformée en une succession d'une courbe et d'un segment de droite (ou en une succession d'un segment de droite et d'une courbe).

2) Une déformation d'un segment de droite : Un segment de droite déformé en une succession d'un segment de droite et d'une courbe (ou en une succession d'une courbe et d'un segment de droite).

Remarque

Les règles établies correspondent au cas où le descripteur global n'est pas modifié ce qui exclut la configuration d'une courbe suivie d'une autre petite courbe bruitée. Le problème dans ce cas est traité par l'algorithme de réduction de descripteurs globaux (voir § 3.3).

Considérons les notations suivantes:

- $\text{Size}(C)$: la longueur de la primitive courbe.
- $\text{Size}(L)$: la longueur de la primitive segment de droite.
- $\text{Size}(C')$: Elle correspond à la longueur de la primitive courbe obtenue après transformation.
- $\text{Size}(L')$: Elle correspond à la longueur de la primitive segment obtenue après transformation.
- $\text{Inclin}(C)$: l'angle d'inclinaison de la primitive courbe selon LWDOS (voir chapitre 2)
- $\text{Inclin}(L)$: l'angle d'inclinaison de la primitive segment de droite.
- $\text{Inclin}(C')$: l'angle d'inclinaison de la primitive obtenue après approximation (Il sera obtenu en fonction de $\text{Inclin}(L)$ et $\text{Inclin}(C)$ comme nous allons voir par la suite).
- $\text{Inclin}(L')$: l'angle d'inclinaison de la primitive segment de droite obtenue après approximation (Il sera obtenu en fonction de $\text{Inclin}(C)$ et $\text{Inclin}(L)$ comme nous allons voir par la suite).

3.4.2.1 Cas d'un segment de droite additionnel

Dans ce cas une courbe est déformée en une succession de deux primitives: un segment de droite suivi d'une courbe (ou une courbe suivie d'un segment). Nous cherchons à inverser le processus afin d'obtenir la description de la courbe initiale à partir de celles des deux primitives [8].

Soit C une courbe suivie par un segment de droite L ou inversement telle que la longueur du segment de droite est négligeable relativement à celle de la courbe et ayant la même orientation. La paire de ces deux primitives sera remplacée par une seule primitive dans le descripteur textuel. Les différents cas sont donnés par le tableau 3.1.

La figure 3.13 montre un exemple d'une succession de deux contours élémentaires (une courbe et un segment de droite). La figure 3.14 montre un exemple d'approximation de deux primitives par une courbe.

- CV $\text{Inclin}(C) \text{ Size}(C), L \text{ Inclin}(L) \text{ Size}(L) \rightarrow \text{CV } \text{Inclin}(C') \text{ Size}(C')$
- CC $\text{Inclin}(C) \text{ Size}(C), L \text{ Inclin}(L) \text{ Size}(L) \rightarrow \text{CC } \text{Inclin}(C') \text{ Size}(C')$

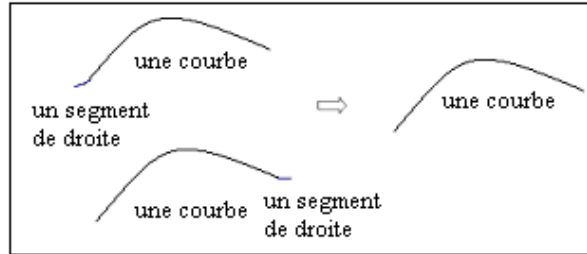


Figure 3.13 Exemples d'approximation par une courbe

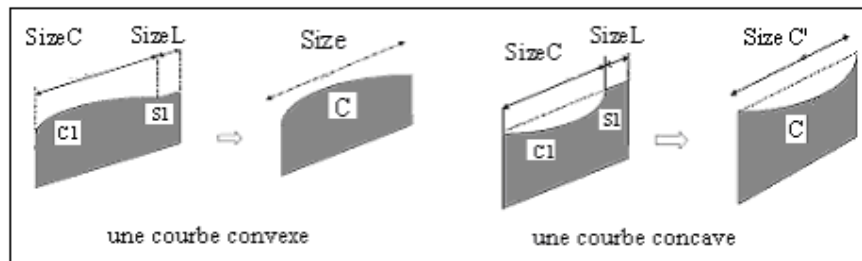


Figure 3.14 Primitive résultat dans le cas d'un segment de droite additionnel

La valeur de $\text{Inclin}(C')$ est donnée par le tableau 3.1 en fonction des inclinaisons de la courbe et du segment ($\text{Inclin}C$ et $\text{Inclin}L$).

$\text{Inclin}C \backslash \text{Inclin}L$	wir	ir	Gir	wil	il	Gil	Horz	Vrt
Wir	wir	wir	-	-	-	-	-	wir
Ir	-	ir	ir	-	-	-	-	wil
Gir	-	-	Gir	-	Vrt	Vrt	-	Gir
Wil	-	-	-	wil	wil	-	-	wil
Il	-	-	-	-	il	il	-	Gil
Gil	-	-	Vrt	-	Vrt	Gil	-	Gil
Horz	-	-	-	-	-	-	Horz	-
Vrt	-	Gir	-	-	Gil	-	-	Vrt

Tableau 3.1 Valeurs de $\text{Inclin}(C')$ pour toutes les configurations possibles

3.4.2.2 Cas d'une courbe additionnelle

En supposant que la longueur de la courbe additionnelle est négligeable relativement à la longueur du segment de droite, et que son orientation est identique à celui du segment de droite, le tableau 3.2 illustre les différents cas possibles.

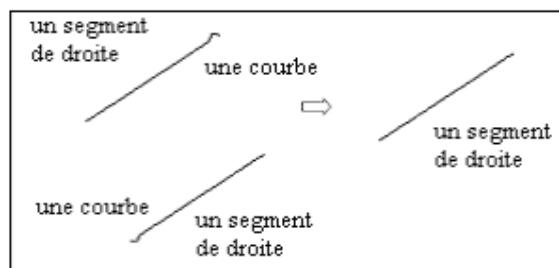


Figure 3.15 Exemples d'approximation par un segment de droite

La figure 3.15 montre un exemple d'une succession de deux contours élémentaires (un segment de droite et une courbe). Les courbes sur les deux contours à gauche de la figure sont très petites, par conséquent ça devient très rigoureux de dire que les silhouettes auxquelles elles appartiennent sont différentes. C'est rigoureux également de dire que les deux contours (la courbe suivie du segment ou inversement) sont différents du contour à droite de la même figure. Vu que les courbes sont très petites il serait acceptable de supposer qu'elles sont dues au bruit pour être plus tolérant dans le processus de comparaison.

La figure 3.16 montre un exemple d'une succession de deux contours élémentaires (un segment de droite et une courbe) et leur approximation par un segment de droite.

- $L \text{ Incl}(L) \text{ Size}(L), CV \text{ Incl}(C) \text{ Size}(C) \rightarrow L \text{ Incl}(L') \text{ Size}(L')$
- $L \text{ Incl}(L) \text{ Size}(L), CC \text{ Incl}(C) \text{ Size}(C) \rightarrow L \text{ Incl}(L') \text{ Size}(L')$

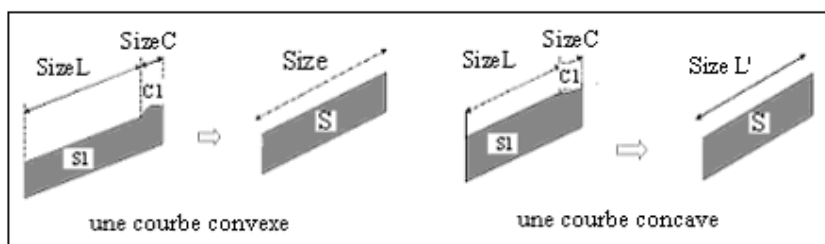


Figure 3.16 Primitive résultat dans le cas d'une courbe additionnelle

Le tableau 3.2 illustre les valeurs de $Inclin(L')$ pour les configurations possibles des inclinaisons de (L) et (C).

InclinC \ InclinL	wir	ir	Gir	wil	il	Gil	Horz	Vrt
wir	wir	-	-	Horz	-	-	-	-
Ir	-	ir	-	-	-	-	-	-
Gir	-	-	Gir	-	-	Vrt	-	-
wil	Horz	-	-	wil	-	-	-	-
Il	-	-	-	-	il	-	-	-
Gil	-	-	Vrt	-	-	Gil	-	-
Horz	-	-	-	-	-	-	Horz	-
Vrt	Wir	wir	Gir	wil	Gil	Gil	-	Vrt

Tableau 3.2 Les valeurs possibles de $Inclin(L')$

Remarque

L'algorithme de transformation de descripteurs détaillés est lui aussi récursif, il peut être appliqué plusieurs fois afin d'atteindre l'approximation désirée (Figure 3.17).

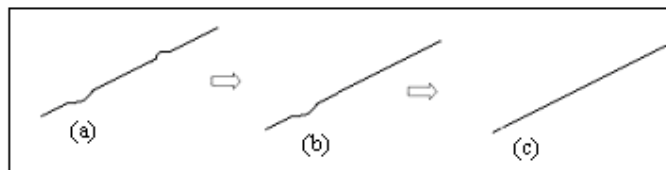


Figure 3.17 Application récursive du processus d'approximation

Le contour (a) dans la figure 3.17 contient des déformations supposées dues au bruit. Les deux déformations seront éliminées en deux étapes comme c'est montré par les contours (b) et (c) de la figure 3.17.

3.5 Validation Expérimentale

3.5.1 Données de l'expérimentation

Nous avons testé notre algorithme sur un ensemble d'images correspondant à :

- Images d'objets 3D isolés (figure 3.18).
- Prises de vues stéréoscopiques d'objets 3D (figure 3.19).
- Images d'objets 3D en mouvement (figure 3.20).

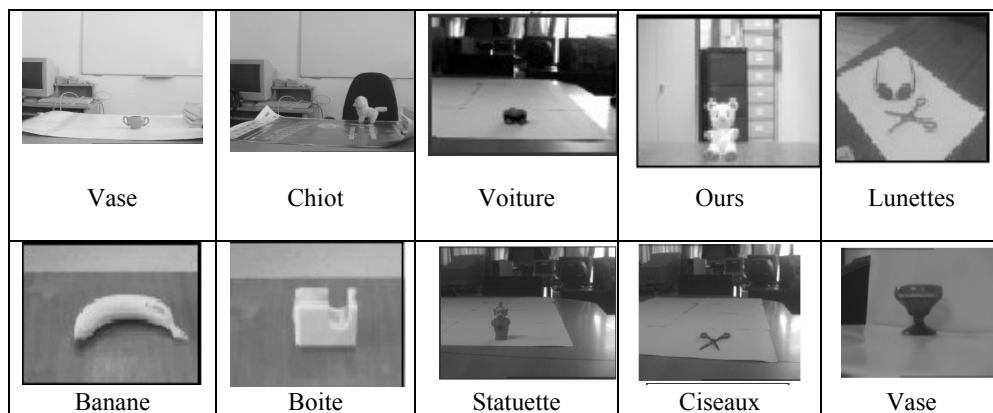


Figure 3.18 Objets utilisés pour l'expérimentation

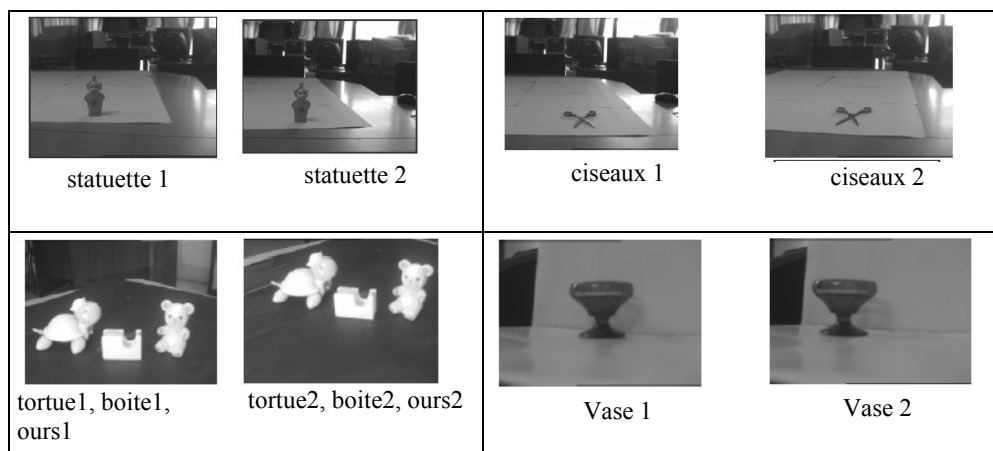


Figure 3.19 Paires d'images stéréoscopiques

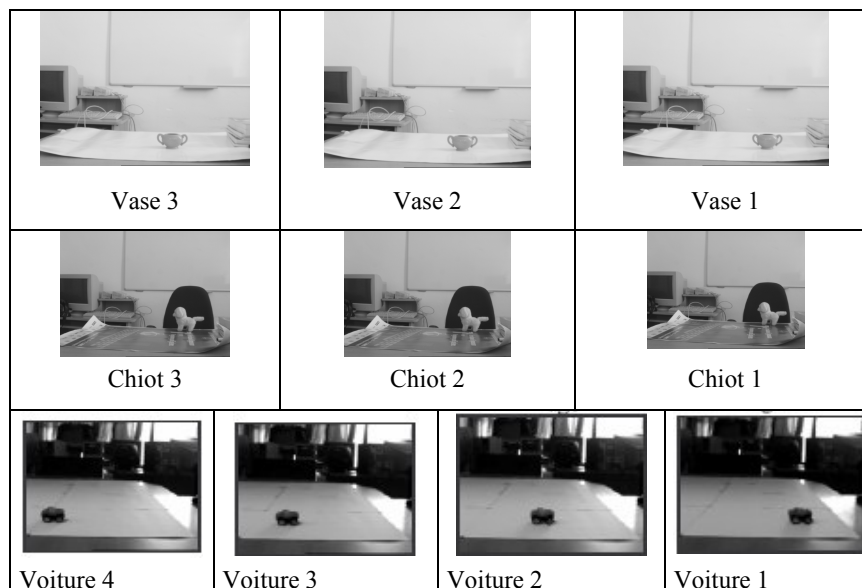


Figure 3.20 Images d'objets en mouvement

3.5.1 Mise en correspondance des descripteurs globaux

La première partie de l'expérimentation consiste à mettre en correspondance les descripteurs globaux de paires de silhouettes en appliquant l'algorithme de décomposition en paires de sous descripteurs.

La taille du bruit est une valeur « `seuil_transf` » pixels. Une partie ayant cette taille est une partie bruitée qui appartient à la silhouette.

Le seuil de transformation « **seuil_transf** » (seuil de tolérance pour la comparaison) est fixé à 5 pixels à la suite de plusieurs expérimentations comme nous l'avons mentionné précédemment. Ce seuil correspond à la différence tolérée sur les deux silhouettes pour les mettre en correspondance.

Nous illustrons quelques résultats obtenus suite à l'application de l'appariement global des descripteurs sur les objets extraits des différentes images.

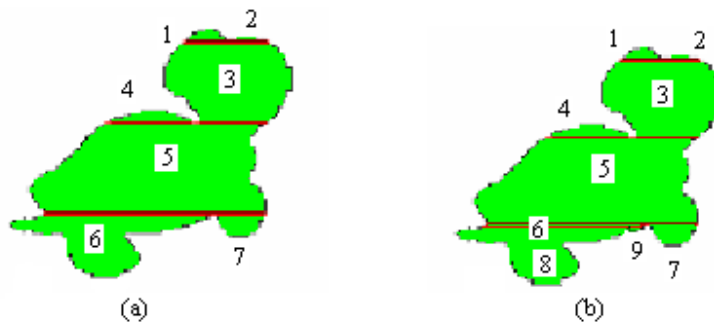


Figure 3.21 Mise en correspondance des descripteurs de tortue1 et tortue2

1- Images stéréoscopique de l'objet tortue (figure 3.19)

Le résultat de la décomposition en parties des silhouettes extraites est illustré par la figure 3.21. Les descripteurs globaux des silhouettes (a) et (b) sont respectivement :

$$(a) \quad [[P1 \cup P2 \oplus \uparrow JL1 \oplus P3] \cup P4 \oplus \uparrow JL2 \oplus P5 \oplus \downarrow DJL1 \oplus P6 \cup P7]$$

$$(b) \quad [[P1 \cup P2 \oplus \uparrow JL1 \oplus P3] \cup P4 \oplus \uparrow JL2 \oplus P5 \oplus \downarrow DJL1 \oplus [P6 \oplus \downarrow DJL2 \oplus P8 \cup P9] \cup P7]$$

Les lignes de séparation appariées sont :

(JL1, JL1) : La description est (Junction Short Junction Short)

(JL2, JL2) : La description est (Junction Less short Junction Less short)

(DJL1, DJL1) : La description est (Junction Enough long Junction Short)

L'algorithme transforme à titre d'exemple la paire $(P6, P6 \oplus \downarrow DJL \oplus P8 \cup P9)$ et donne $(P6, P'6)$ et ceci après avoir effectué la réduction: $P6 \oplus \downarrow DJL \oplus P8 \cup P9 \rightarrow P'6$. Ainsi, les paires de descripteurs obtenues sont :

$$(P1, P1), (P2, P2), (P3, P3), (P4, P4), (P5, P5), (P6, P6 \oplus \downarrow DJL \oplus P8 \cup P9), (P7, P7)$$

Après application de l'algorithme, les descripteurs globaux des deux silhouettes deviennent:

$$(a) \quad [[P1 \cup P2 \oplus \uparrow JL1 \oplus P3] \cup P4 \oplus \uparrow JL2 \oplus P5 \oplus \downarrow DJL1 \oplus P6 \cup P7]$$

$$(b) \quad [[P1 \cup P2 \oplus \uparrow JL1 \oplus P3] \cup P4 \oplus \uparrow JL2 \oplus P5 \oplus \downarrow DJL1 \oplus P'6 \cup P7]$$

Ces descripteurs sont donc identiques et sont donc sujet à un appariement détaillé.

2- Images stéréoscopiques d'un objet (ours) (voir figure 3.19) et dont le résultat de leur décomposition en parties est illustré par la figure 3.22 est donné par :

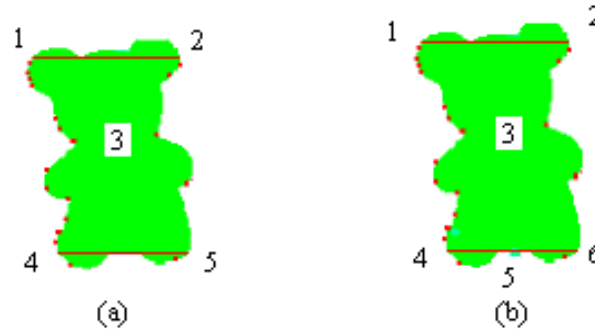


Figure 3.22 Mise en correspondance des descripteurs de Ours1 et Ours2

Les descripteurs globaux des silhouettes (a) et (b) sont respectivement :

(a) $[P1 \cup P2 \oplus \uparrow JL1 \oplus P3 \oplus \downarrow DJL1 \oplus P4 \cup P5]$

(b) $[P'1 \cup P2 \oplus \uparrow JL1 \oplus P3 \oplus \downarrow DJL1 \oplus P4 \cup P5 \cup P6]$

Les lignes de séparation appariées sont :

(JL1, JL1) : La description est (Junction Less short Junction Enough long)

Les paires de descripteurs obtenues sont :

(P1, P1), (P2, P2), (P3 \oplus \downarrow DJL1 \oplus P4 \cup P5, P3 \oplus \downarrow DJL1 \oplus P4 \cup P5 \cup P6)

Après application de l'algorithme, les descripteurs globaux des deux silhouettes deviennent:

(a) $[P1 \cup P2 \oplus \uparrow JL1 \oplus P3 \oplus \downarrow DJL1 \oplus P4 \cup P5]$

(b) $[P1 \cup P2 \oplus \uparrow JL1 \oplus P3 \oplus \downarrow DJL1 \oplus P4 \cup P5]$

Ces descripteurs sont donc identiques et sont donc sujet à un appariement détaillé.

3.5.2 Mise en correspondance des descripteurs détaillés

Après la correction des descripteurs globaux et élimination des parties bruitées, nous appliquons les règles d'approximation pour la correction des descripteurs détaillés et la mise en correspondance de deux silhouettes.

La détermination des descripteurs détaillés nécessite la localisation des points d'inflexion sur le contour de la silhouette. Pour ce faire LWDOS utilise un algorithme connu de chetverikof [50]. Trois exemples sont présentés ci dessous pour lesquels les descripteurs détaillés deviennent similaires après l'application des règles d'approximation. Les sous descripteurs soulignés sont ceux concernés par les transformations.

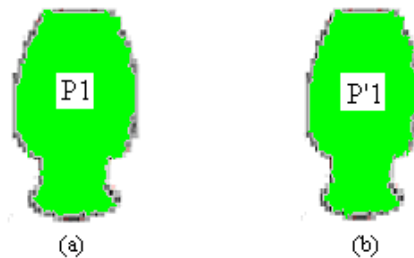


Figure 3.32 Mise en correspondance des descripteurs détaillés des silhouettes d'un vase

- Chacune des silhouettes de la figure 3.32 est composée d'une seule partie. Le descripteur global de la silhouette (a) est [P1] et celui de la silhouette (b) est [P'1].

Nous donnons maintenant les descripteurs détaillés de P1 et P'1 :

[P1] : {Ln Hoz SH, wev wil VS, Ln Vrt VS, wev Gil ES, Ln Vrt VS, wev ir ES, Ln ir VS, Ln Vrt VS, Ln Vrt VS, wev il VS, wev ir VS, Ln Hoz ES ⊗ Ln Hoz SH, wev wir VS, Ln Vrt VS, wev Gir ES, Ln Vrt VS, wev Gil ES, Ln il VS, Ln Vrt VS, wev ir VS, wev il SH, Ln Hoz ES}

[P'1] : {Ln Hoz SH, wev wil VS, Ln Vrt VS, wev Gil ES, Ln Vrt VS, wev ir SH, Ln ir VS, Ln Vrt VS, wev il VS, wev ir VS, Ln Hoz ES ⊗ Ln Hoz SH, wev wir VS, Ln Vrt VS, wev Gir ES, Ln Vrt VS, wev Gil ES, Ln il VS, Ln Vrt VS, wev ir VS, wev il SH, Ln Hoz ES}

En analysant ces deux descripteurs nous remarquons qu'ils sont différents. Nous appliquons alors une liste de transformations. Les descripteurs soulignés sont ceux concernés par les approximations, ce sont ceux qui ont rendu les descripteurs de P1 et P'1 différents.

Nous appliquons les transformations ci-dessous :

Pour la silhouette (a): $wcv\ ir\ sizec + Ln\ ir\ sizer \rightarrow wcv\ ir\ size$
 $wcv\ ir\ sizec + Ln\ vrt\ sizer \rightarrow wcv\ ir\ size$

Pour la silhouette (b) : $wcv\ ir\ sizec + Ln\ ir\ sizer \rightarrow wcv\ ir\ size$

En remplaçant dans les descripteurs de (a) et (b) les parties gauches des règles d'approximation par les parties droites, nous obtenons le même descripteur détaillé de P1 et P'1 qui est :

$[P1] = [P'1] = \{Ln\ Hoz\ SH, wcv\ wil\ VS, Ln\ Vrt\ VS, wcv\ Gil\ ES, Ln\ Vrt\ VS, wcv\ ir\ ES, Ln\ Vrt\ VS, wcv\ il\ VS, wcv\ ir\ VS, Ln\ Hoz\ ES \otimes Ln\ Hoz\ SH, wcv\ wir\ VS, Ln\ Vrt\ VS, wcv\ Gir\ ES, Ln\ Vrt\ VS, wcv\ Gil\ ES, Ln\ il\ VS, Ln\ Vrt\ VS, wcv\ ir\ VS, wcv\ il\ SH, Ln\ Hoz\ ES\}$

- Les silhouettes (a) et (b) de la figure 3.33 ont le même descripteur global conformément à LWDOS, il est de la forme : $[P1UP2 \oplus \uparrow JL1 \oplus P3 \oplus \downarrow DJL1 \oplus P4UP5]$.

Les lignes de jonctions JL1 et JL'1 des silhouettes (a) et (b) sont mises en correspondance, elles ont le même descripteur donné par : « Junction Middle, Free Low Very Short, Junction Short ».

De même pour la disjonction DJL1 et DJL'1 est : « Junction Short, Free Heigh Very Short, Junction Middle ».

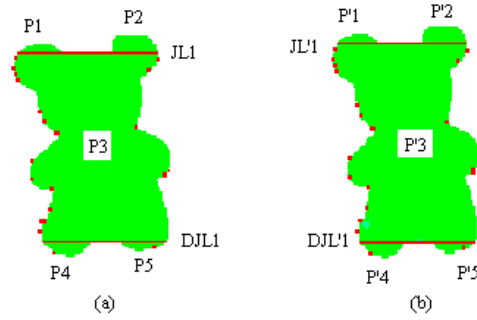


Figure 3.33 Mise en correspondance des descripteurs détaillés des silhouettes d'un Ours

Les descripteurs de chaque partie de la silhouette (a) est donnée par :

[P1] : { wcv wil SH, Ln Vrt VS \otimes wcv ir SH }

[P2] : { Ln Hoz VS, wcv il VS \otimes Ln Hoz VS, wcv wir VS }

[P3] : { wcv ir SH, wcv Gir VS, Ln Vrt VS, wcv ir VS, wcv wil SH, wcv ir SH, Ln Hoz VS, wcv il ES, wcv wir VS \otimes wcv ir VS, wcv il VS, wcv Gil SH, wcv ir VS, **Ln Vrt SH, wcv il VS, Ln Vrt VS, wcv Gil ES** }

[P4] : { Ln ir VS \otimes wcv wil VS }

[P5] : { wcv ir VS, wcv ir VS, Ln Hoz SH \otimes wcv Gil SH, Ln Hoz SH }

De même les descripteurs de chaque partie de la silhouette (b) est donnée par :

[P'1] : { wcv wil SH, Ln Vrt VS \otimes wcv ir SH }

[P'2] : { Ln Hoz VS, wcv il VS \otimes Ln Hoz VS, wcv wir VS }

[P'3] : { wcv ir SH, wcv Gir VS, Ln Vrt VS, wcv ir VS, wcv wil SH, Ln **Vrt VS, wcv ir SH**, Ln Hoz VS, **wcv il SH, Ln Vrt VS**, wcv wir VS \otimes wcv ir VS, wcv il VS, wcv Gil SH, wcv ir VS, Ln Gil SH, wcv Gil ES }

[P'4] : { Ln ir VS \otimes wcv wil VS }

[P'5] : { wcv ir VS, wcv ir VS, Ln Hoz SH \otimes wcv Gil SH, Ln Hoz SH }

Comme les descripteurs détaillés sont différents, nous appliquons les transformations suivantes :

Pour la silhouette (a) : Ln vrt sizel + wcv il size \rightarrow wcv Gil size

Ln vrt sizel + wcv Gil size \rightarrow wcv Gil size

Pour la silhouette (b) : $Ln\ vrt\ size\ +\ wcv\ ir\ size\ \rightarrow\ wcv\ ir\ size\ c$

$wcv\ il\ size\ c\ +\ Ln\ vrt\ size\ \rightarrow\ wcv\ il\ size$

Après transformation nous obtenons le même descripteur pour les deux silhouettes donné par:

$[P1] = [P'1] : \{ wcv\ wil\ SH, Ln\ Vrt\ VS \otimes\ wcv\ ir\ SH \}$

$[P2] = [P'2] : \{ Ln\ Hoz\ VS, wcv\ il\ VS \otimes\ Ln\ Hoz\ VS, wcv\ wir\ VS \}$

$[P3] = [P'3] : \{ wcv\ ir\ SH, wcv\ Gir\ VS, Ln\ Vrt\ VS, wcv\ ir\ VS, wcv\ wil\ SH, wcv\ ir\ SH, Ln\ Hoz\ VS, wcv\ il\ ES, wcv\ wir\ VS \otimes\ wcv\ ir\ VS, wcv\ il\ VS, wcv\ Gil\ SH, wcv\ ir\ VS, Ln\ Gil\ SH, wcv\ Gil\ ES \}$

$[P4] = [P'4] : \{ Ln\ ir\ VS \otimes\ wcv\ wil\ VS \}$

$[P5] = [P'5] : \{ wcv\ ir\ VS, wcv\ ir\ VS, Ln\ Hoz\ SH \otimes\ wcv\ Gil\ SH, Ln\ Hoz\ SH \}$

- Le descripteur global des silhouettes (a) et (b) de la figure 3.34 est de la forme:

$[P1UP2 \oplus \uparrow JL1 \oplus P3UP4 \oplus \uparrow JL2 \oplus P5 \oplus \downarrow DLJ1 \oplus [P6 \oplus \downarrow DLJ2 \oplus P8UP9UP10]UP7]$.

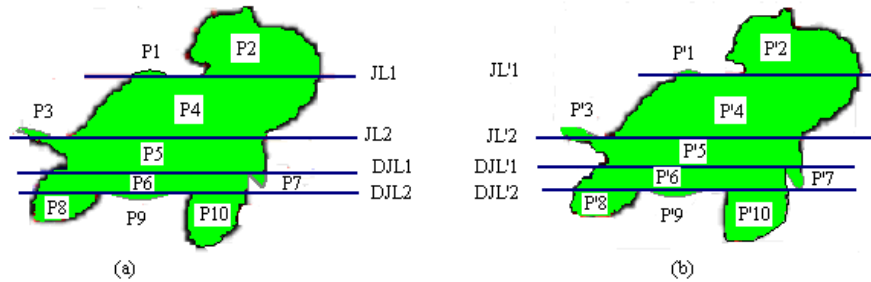


Figure 3.34 Mise en correspondance des descripteurs détaillés des silhouettes d'une tortue

Les deux lignes de jonction nommées JL1 et JL'1 sur les deux silhouettes sont mises en correspondance. Leur descripteur est : "Junction Very Short, Junction Enough Short".

Les deux lignes de jonction nommées JL2 et JL'2 sur les deux silhouettes sont mises en correspondance. Leur descripteur est : "Junction Very Short, Free Low Very Short, Junction Enough Long".

Les deux lignes de disjonction nommées DJL1 et DJL'1 sur les deux silhouettes sont mises en correspondance. Leur descripteur est : " Junction Enough Long, Junction Very Short".

Les deux lignes de disjonction nommées DJL2 et DJL'2 sur les deux silhouettes sont mises en correspondance. Leur descripteur est: " Junction Enough Short, Junction Enough Short, Junction Enough Short ".

Le descripteur détaillé de la silhouette (a) est:

- [P1] : { Ln Hoz SH, wev wil VS \otimes Ln Hoz SH, wev, wir VS }
- [P2] : { Ln Hoz VS, Ln il VS, Ln Hoz VS, wev il SH, Ln il VS, Ln Vrt VS, Ln wir VS, Ln Vrt SH, wcc
Vrt VS \otimes Ln Hoz VS, Ln ir VS, Ln Hoz VS, wev ir SH, Ln Vrt VS }
- [P3] : { Ln Hoz VS, wev Gir VS \otimes Ln Hoz VS, wev wir VS }
- [P4] : { Ln il VS, wev il SH, Ln il SH, Ln Vrt VS \otimes wev il SH, wcc il VS }
- [P5] : { Ln ir VS, Ln Hoz VS, wev wir VS, Ln il VS \otimes Ln Vrt VS }
- [P6] : { Ln il VS \otimes wev ir VS }
- [P8] : { Ln Gil VS, Ln Vrt VS, Ln ir VS, Ln Hoz SH \otimes wev il SH, Ln Hoz SH }
- [P9] : { wev wir VS, Ln Hoz SH \otimes wev wil VS, Ln Hoz SH }
- [P10] : { Ln Vrt SH, Ln ir VS, Ln Hoz VS \otimes Ln Vrt VS, wev il SH, Ln Hoz VS }
- [P7] : { Ln ir VS \otimes wev Gil VS }

Le descripteur détaillé de la silhouette (b) est:

- [P'1] : { Ln Hoz SH, wev wil VS \otimes Ln Hoz SH, wev, wir VS }
- [P'2] : { Ln Hoz VS, Ln il VS, Ln Hoz VS, wev il SH, Ln il VS, Ln Vrt VS, Ln wir VS, Ln Vrt SH, wcc
Vrt VS \otimes Ln Hoz VS, Ln ir VS, Ln Hoz VS, wev ir SH, Ln Vrt VS }
- [P'3] : { Ln Hoz VS, wev Gir VS \otimes Ln Hoz VS, wev wir VS }
- [P'4] : { Ln il VS, wev il SH, Ln il SH, Ln Vrt VS \otimes wev il SH, wcc il VS }
- [P'5] : { Ln ir VS, Ln Hoz VS, wev wir VS, Ln il VS \otimes Ln Vrt VS }
- [P'6] : { Ln il VS \otimes wev ir VS }
- [P'8] : { Ln Gil VS, Ln Vrt VS, Ln ir VS, Ln Hoz SH \otimes wev il SH, Ln Hoz SH }
- [P'9] : { wev wir VS, Ln Hoz SH \otimes wev wil VS, Ln Hoz SH }
- [P'10] : { Ln Vrt SH, Ln ir VS, Ln Hoz VS \otimes Ln Vrt VS, wev il SH, Ln Hoz VS }
- [P'7] : { Ln ir VS \otimes wev Gil VS }

Comme les descripteurs détaillés sont différents, nous appliquons les transformations suivantes :

Pour la silhouette (a) : $wcv\ il\ sizec + Ln\ il\ sizel \rightarrow wcv\ il\ size$
 $Ln\ vrt\ sizel + wcv\ vrt\ sizec \rightarrow Ln\ vrt\ size$

Pour la silhouette (b) : $wcv\ il\ sizec + Ln\ vrt\ sizel \rightarrow wcv\ il\ size$

Après transformation nous obtenons le même descripteur pour les deux silhouettes donné par:

[P1] = [P'1] : { Ln Hoz SH, wcv wil VS \otimes Ln Hoz SH, wcv, wir VS }

[P2] = [P'2]: { Ln Hoz VS, Ln il VS, Ln Hoz VS, wcv il ES, Ln wir VS, Ln Vrt ES \otimes Ln Hoz VS, Ln ir VS, Ln Hoz VS, wcv ir SH, Ln Vrt VS }

[P3] = [P'3]: { Ln Hoz VS, wcv Gir VS \otimes Ln Hoz VS, wcv wir VS }

[P4] = [P'4]: { Ln il VS, wcv il SH, Ln il SH, Ln Vrt VS \otimes wcv il SH, wcc il VS }

[P5] = [P'5]: { Ln ir VS, Ln Hoz VS, wcv wir VS, Ln il VS \otimes Ln Vrt VS }

[P6] = [P'6]: { Ln il VS \otimes wcv ir VS }

[P8] = [P'6]: { Ln Gil VS, Ln Vrt VS, Ln ir VS, Ln Hoz SH \otimes wcv il SH, Ln Hoz SH }

[P9] = [P'9]: { wcv wir VS, Ln Hoz SH \otimes wcv wil VS, Ln Hoz SH }

[P10] = [P'10]: { Ln Vrt SH, Ln ir VS, Ln Hoz VS \otimes Ln Vrt VS, wcv il SH, Ln Hoz VS }

[P7] = [P'7]: { Ln ir VS \otimes wcv Gil VS }

Discussion et critiques

Nous remarquons à partir des expérimentations effectuées que la mise en correspondance des descripteurs textuels globaux et détaillés est possible en présence du bruit.

Cependant la méthode présentée se base sur la version LWDOS de la description textuelle qui permet d'avoir une description relative et par conséquent les approximations sont aussi exprimées grossièrement. L'application de cette méthode peut être utilisée pour la comparaison grossière des formes ainsi que le suivi d'objets en mouvement, cependant elle n'est pas facilement applicable pour la reconnaissance des formes exigeant l'utilisation des valeurs exactes des attributs des images, telles que les longueurs et les angles.

3.6 Conclusion

Nous avons présenté dans ce chapitre une méthode de mise en correspondance structurelle et sémantique de descripteurs textuels de silhouettes. Nous avons commencé par l'appariement des descripteurs globaux et ce en développant un algorithme de transformation de descripteurs qui garantit un niveau de souplesse et de tolérance dans le processus de mise en correspondance. Ensuite nous nous sommes intéressés à l'approximation des descripteurs détaillés en appliquant un ensemble de règles définies.

Les résultats sont satisfaisants mais ils sont limités à une comparaison grossière des formes car LWDOS décrit les formes avec leurs dimensions relatives (un pourcentage par rapport au rectangle d'aire minimum). On peut appliquer cette méthode dans le cadre de la vision et de la reconnaissance grossière des formes mais non pas dans des applications multimédia nécessitant de connaître les dimensions exactes. Le moins qu'on puisse tirer des expérimentations effectuées est que la présence du bruit peut être ignorée ou plutôt tolérée dans le processus de comparaison. Les résultats obtenus nous ont encouragé à entamer la reconnaissance avec le langage XLWDOS qui lui, s'intéresse aux dimensions exactes donc ayant un champ d'applications plus large que son prédécesseur.

Notre objectif consiste à reconnaître les silhouettes d'objets 3D. Il s'agira en fait de mettre en correspondance une silhouette requête (de l'objet qu'on veut reconnaître) avec des silhouettes modèles dont les descriptions sont stockées en mémoire. Il sera donc indispensable de passer par une étape d'indexation et de classification des silhouettes modèles et ce pour garantir une rapidité dans le processus de reconnaissance.

Le chapitre suivant est consacré à la reconnaissance des silhouettes d'objets en utilisant la description textuelle XLWDOS.

Chapitre 4

Reconnaissance des descripteurs textuels de silhouettes

4.1 Introduction

L'indexation et la recherche d'images dans des bases d'images sont des sujets de grand intérêt pour la vision artificielle et la reconnaissance des formes.

Nous proposons dans ce chapitre une méthode de reconnaissance d'objets permettant la mise en correspondance entre une silhouette requête et des silhouettes modèles. Les silhouettes modèles sont stockées en mémoire en utilisant leurs descriptions textuelles écrites dans le langage XLWDOS [121, 120, 117].

Nous nous intéressons en premier lieu à l'effet de la déformation de la frontière d'une silhouette sur son descripteur textuel. Nous proposons ensuite un schéma pour l'organisation de la base de descripteurs de silhouettes modèles et leur indexation.

En second lieu nous proposons deux méthodes pour résoudre le problème de la reconnaissance d'une silhouette: la première concerne la recherche des silhouettes candidates à la similarité pour une silhouette requête. La seconde concerne la mesure de similarité entre la silhouette requête et les silhouettes modèles afin de sélectionner le meilleur résultat.

4.2 La description XLWDOS

Nous rappelons brièvement dans ce paragraphe l'écriture de la description d'une forme par le langage XLWDOS présenté au chapitre 2. .

Le langage XLWDOS [121] est un langage XML développé pour la description des silhouettes, il est issu directement du langage LWDOS [120]. L'exemple de la figure 4.1 illustre brièvement la description d'une silhouette moyennant le langage XLWDOS [121, 120, 117]. Le descripteur global XLWDOS est donné par :

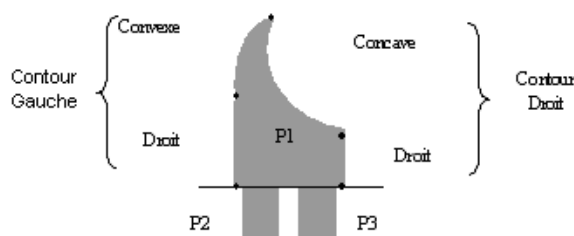


Figure 4.1 Exemple de description d'une silhouette

```
<DS><nom> exemple de silhouette</nom>
```

```
<CP> P1 D1 P2 P3 </CP></DS> (DS signifie description of shape).
```

Cette description signifie que la silhouette est formée d'une partie P1 suivie d'une ligne de disjonction (D1) suivi à son tour de deux autres parties P2 et P3. Il est à noter que lorsqu'on passe de plusieurs parties à une seule, la ligne sera appelée ligne de jonction (J). Comme la même description globale peut décrire plusieurs silhouettes d'objets, il serait nécessaire de passer à la description détaillée des éléments.

Le descripteur détaillé est donné par :

```
<DS><nom> exemple de silhouette</nom>
```

```
<CP>
```

```
<P1><L> cv 3 130° 25 r 90° 35</L><R> cc 25 60° 40 r 90° 20</R></P1>
```

```
<D1> h P1 5 s P1 P2 15 h P1 9 s P1 P3 17 h P1 6 </D1>
```

```
<P2><L> r 90 20 r 180 15 </L><R> r 90 20 </R></P2>
```

<P3><L> r 90 20 r 180 15 </L><R> r 90 20 </R></P3></CP></DS>.

La désignation CP signifie partie composée (Composed Part).

A titre d'exemple Le côté gauche de la partie P1 est une succession d'une courbe convexe (Cv) dont le degré de convexité est de 3%, l'angle d'inclinaison est de 130° et de longueur 25 pixels. Cette courbe est suivie d'un segment de droite (r) incliné de 90° et de longueur égale à 35 pixels.

Le côté droit de P1 est une courbe concave (Cc) suivi d'un segment vertical.

La ligne de disjonction (D) est constituée d'une succession de segments :

h P1 5 : de longueur 5 pixels, jointe à la partie P1 d'en haut d'où le nom free high symbolisé par (h)

s P1 P2 15 : de longueur 15 pixels jointe aux parties P1 et P2 d'où le nom « shared segment » symbolisé par (s)

h P1 9 : segment de type « h » de longueur 9 pixels appartenant à la partie P1

s P1 P3 17 : segment de type « s » de longueur 17 pixels appartenant à deux parties P1 et P3.

h P1 6 : segment de type « h » de longueur 6 pixels appartenant à la partie P1.

Il est important d'indiquer que toutes ces valeurs et mesures sont calculées par rapport à l'unique rectangle d'aire minimum englobant la silhouette [80].

4.2.1 Effet du bruit sur le descripteur XLWDOS

Toute modification de l'allure des contours d'une forme se répercute directement sur leurs descripteurs. A titre d'exemple, la figure 4.3 représente la même forme que celle de la figure 4.2 avec quelques différences sur le contour due à quelques pixels de plus. Malgré que la différence ne soit pas importante, les deux descripteurs XLWDOS ne sont pas les mêmes.

La différence ne se situe pas seulement au niveau des détails (la géométrie des parties), mais elle concerne aussi leurs structures.

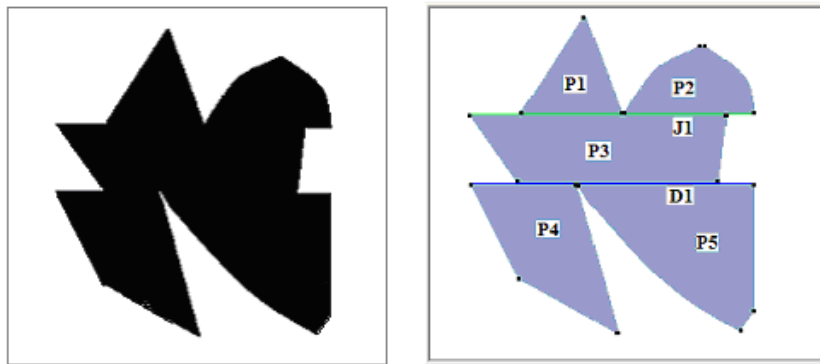


Figure 4.2 Exemple de silhouette et sa décomposition en parties

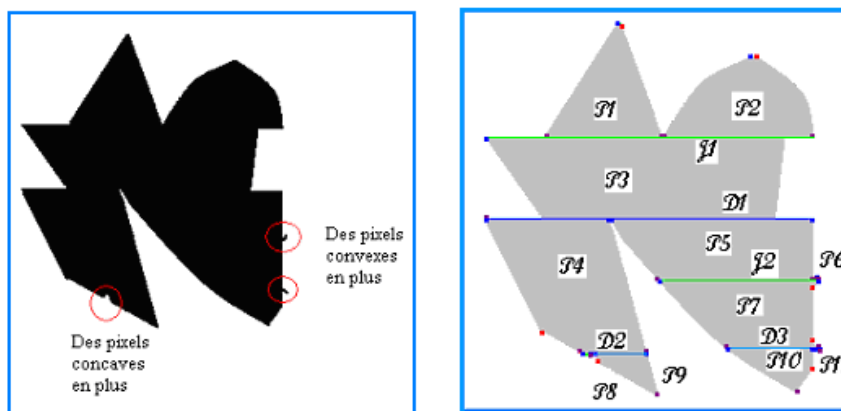


Figure 4.3 Déformation des frontières de la silhouette exemple et résultat de sa décomposition en parties.

En effet le descripteur global XLWDOS de la silhouette de la figure 4.3 est :
 $\langle CP \rangle \langle CP \rangle P_1 P_2 J_1 P_3 \langle /CP \rangle D_1 \langle CP \rangle P_4 D_2 P_8 P_9 \langle /CP \rangle \langle CP \rangle \langle CP \rangle P_5 P_6 J_2 P_7 \langle /CP \rangle D_3 P_{10} P_{11} \langle /CP \rangle \langle /CP \rangle$ alors que celui de la figure 4.2 est :

$\langle CP \rangle \langle CP \rangle P_1 P_2 J_1 P_3 \langle /CP \rangle D_1 P_4 P_5 \langle /CP \rangle$

Nous présentons dans ce qui suit notre algorithme de reconnaissance qui tient compte de ce problème d'altération de descripteurs à cause du bruit.

4.3 Reconnaissance globale de silhouettes

4.3.1 Principe de base

Nous disposons d'une base de descripteurs de silhouettes modèles. La première étape consiste à indexer ces descripteurs afin de faciliter la recherche.

La seconde étape consiste, à partir d'une silhouette requête, de trouver dans la base des descripteurs le descripteur modèle correspondant. Tout d'abord, le descripteur textuel est calculé.

- **Lissage de la silhouette requête** : La silhouette requête est lissée dans le but d'avoir la même structure que les descripteurs des silhouettes modèles.
- **Comparer les nouveaux descripteurs XLWDOS** : Le nouveau descripteur requête qui résulte du lissage est à nouveau comparé avec les descripteurs modèles.
- **Mesure de similarité des descripteurs globaux** : les descripteurs qui sont globalement similaires et ayant donc la même structure que celui de la requête seront considérés pour leur appliquer les mesures de similarité des descripteurs globaux et ce pour retrouver les modèles qui ressemblent le plus à la requête.
- **Comparaison des descripteurs détaillés** : La mesure de similarité utilisée pour les descripteurs globaux nécessitera de passer aux comparaisons des descripteurs détaillés pour compléter la reconnaissance.

4.3.2 Mise en œuvre de la base de descripteurs de silhouettes modèles

4.3.2.1 Calcul des descripteurs de silhouettes

Pour minimiser le volume d'information à stocker on doit générer la description textuelle de chaque image en appliquant le générateur automatique des descripteurs XLWDOS. Une décomposition de la silhouette en parties, lignes de jonction et de disjonction est alors opérée générant un descripteur global de la forme.

La seconde étape de la méthode consiste à calculer les détails de chaque élément du descripteur global. Pour cela, les points de courbure seront localisés sur le contour de chaque partie. Pour détecter les points de courbure l'algorithme de Chetvericof [50] est utilisé.

Le calcul des descripteurs XLWDOS des deux images permet de détecter la présence des parties secondaires (supposées dues au bruit) et cela en calculant la taille de chaque partie [120, 121, 120, 117]. Le bruit dans notre approche correspond à toute partie secondaire dont la taille est inférieure à un seuil donné.

Une fois les descripteurs XLWDOS des silhouettes calculés, ils sont soumis à l'étape d'indexation.

4.3.2.2 Indexation des descripteurs

L'indexation permet d'effectuer une comparaison grossière entre la requête et les modèles et ce dans le but de réduire considérablement le nombre d'images modèles pour la suite du traitement. En effet, ce sont les résultats de cette étape qui seront concernés par des comparaisons ultérieures. Les descripteurs modèles à garder pour l'étape suivante seront ceux qui ont le même index que la requête.

L'index que nous proposons pour un descripteur XLWDOS (permettant de différencier deux descripteurs) est composé des données suivantes [8, 13]:

- Le nombre de parties
- Le nombre de lignes de jonction
- Le nombre de lignes de disjonction
- L'ordre des lignes représenté par une liste de valeurs '0' et '1' tel que '0' correspond à la ligne de jonction et '1' à la ligne de disjonction.
- Le nombre de parties attachées à chaque ligne de séparation (jonction ou disjonction) représenté par une liste de valeurs V_i tel que V_i correspond au nombre de parties associées à la ligne numéro i .
- Les dispositions des parties relativement à chaque ligne et représentées par une liste de caractères 's', 'h', 'w' indiquant les dispositions de toutes les parties

attachées. ('s' pour "junction", 'h' pour "free high" et 'w' pour "free low"). Un caractère spécial (&) est inséré pour séparer les descriptions de deux lignes.

Par exemple l'index du descripteur textuel de la silhouette représentée par la figure 4.4 est (5, 1, 1, 01, 3, 3, wssh&wssw) où:

- **5** est le nombre de parties.
- **1** est le nombre de lignes de jonction
- **1** est le nombre de lignes de disjonction.
- **01** indique l'ordre des lignes de séparation : une ligne de jonction est suivie par une ligne de disjonction.
- **3,3** indique que trois parties sont reliées à la ligne de jonction et trois autres à la ligne de disjonction.
- **wssh&wssw** indique que la première ligne est composée de quatre segments ayant les attributs : w, s, s, h et la deuxième est composée de trois segments ayant les attributs: w, s, s, w.

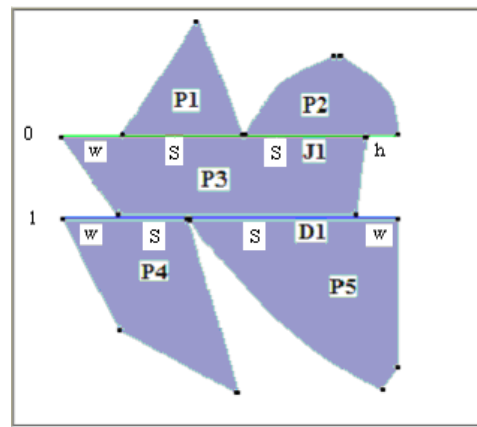


Figure 4.4 Indexation d'une forme

Plusieurs formes peuvent avoir le même index, la différence sera recherchée au niveau de la géométrie des parties et au niveau des longueurs des segments des lignes de séparation de chaque forme.

4.3.3 Recherche d'une silhouette requête

4.3.3.1 La position du problème

En supposant que toutes les descriptions textuelles des silhouettes modèles sont stockées et indexées, une condition nécessaire pour qu'une silhouette requête soit mise en correspondance avec une silhouette modèle est que leurs descriptions XLWDOS aient le même index.

Etant donnée que la description utilisée dans ce travail est sensible au bruit, alors deux formes similaires risquent d'avoir des index différents. D'autres parties ou lignes peuvent apparaître sur les formes, par conséquent les descriptions de ces dernières marqueront leur présence dans la description de la forme.

Les silhouettes (a) et (b) de la figure 4.5 illustrent des exemples de la déformation produisant d'autres parties et lignes. Les parties supposées dues au bruit sont caractérisées par des longueurs négligeables relativement aux tailles des autres parties de la forme.

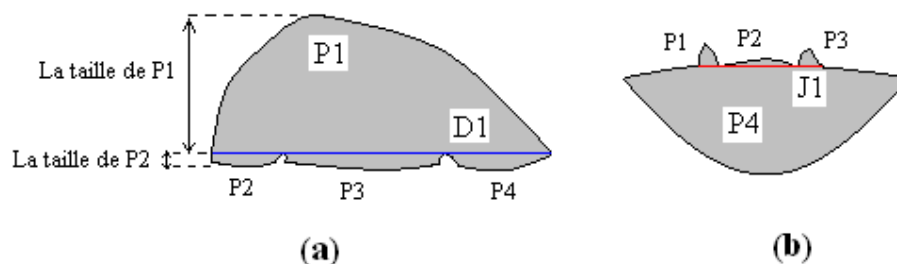


Figure 4.5 Exemples de parties additionnelles dues au bruit

Dans le cas réel, les silhouettes extraites sont souvent entachées d'un bruit, et la recherche de l'index calculé dans la table des index de la base de descripteurs peut échouer en raison des différences dans la structure du descripteur requête produites par le bruit.

Dans le cas où cet index calculé n'est pas retrouvé comme tel, il sera judicieux d'éliminer du descripteur requête toute information additionnelle générée par un bruit.

Nous proposons deux méthodes pour résoudre ce problème :

- La première consiste à appliquer des transformations au descripteur requête dans le but de trouver un sous ensemble de descripteurs modèles ayant le même index

- La seconde consiste à appliquer un lissage de la frontière de la forme requête afin d'éliminer le bruit qui empêche d'obtenir un index identique dans la table des index.

4.3.3.2 Méthode de transformation du descripteur de la silhouette requête

Rappelons les définitions suivantes abordées dans le chapitre 3 concernant la mise en correspondance selon LWDOS [120]:

- **Une partie secondaire** comme étant la partie qui apparaît à gauche d'une ligne de jonction (avant la ligne de jonction dans le descripteur XLWDOS) ou à droite d'une ligne de disjonction (après la ligne de disjonction dans le descripteur XLWDOS).
- **Une partie principale** comme étant la partie qui apparaît à droite d'une ligne de jonction dans le descripteur XLWDOS ou à gauche d'une ligne de disjonction dans le descripteur XLWDOS.

Lorsque la taille d'une partie secondaire (P_{si}) dans une partie composée est négligeable relativement à la taille de la partie principale (inférieure à un seuil $LONG_{SEUIL}$) alors la partie composée associée à P_{si} va être transformée ou réduite comme suit :

Proposition 1 :

Soient :

- $\langle CP \rangle P_{s1} P_{s2} \dots P_{s(i-1)} P_{si} \dots P_{sn} J_i P_m \langle /CP \rangle$ une partie composée du descripteur requête.
- La partie secondaire (P_{si}) est de longueur inférieure au seuil $LONG_{SEUIL}$.

La partie (P_{si}) est alors considérée issue d'un bruit et sera donc éliminée de l'expression de la partie composée. La nouvelle partie composée devient :

$$\langle CP \rangle P_{s1} P_{s2} \dots P'_{s(i-1)} P_{s(i+1)} \dots P_{sn} J'_i P_m \langle /CP \rangle$$

La partie $P_{s(i-1)}$ devient donc $P'_{s(i-1)}$ tel que la frontière gauche de $P'_{s(i-1)}$ est identique à celle de $P_{s(i-1)}$ alors que son contour droit est celui de $P_{s(i-1)}$ augmenté par le contour de (P_{si}) ainsi que les segments de la ligne de séparation en relation avec P_{si} et ayant les attributs w ($free_low$) (voir figure 4.6).

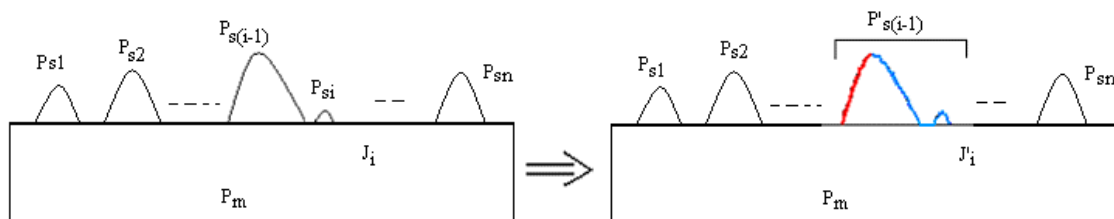


Figure 4.6 Exemple d'élimination d'une partie bruitée

La description de la ligne de jonction J_i doit aussi changer et devient J'_i où les segments associés à la partie P_{si} dans la ligne J_i doivent être attribués à la nouvelle partie $P'_{s(i-1)}$.

Dans le cas d'absence de parties à gauche de P_{si} alors le contour de P_{si} deviendra la continuation de la frontière gauche de la partie $P'_{s(i+1)}$ et les segments associés à la partie P_{si} dans la ligne J_i doivent être attribués à la nouvelle partie $P'_{s(i+1)}$ (voir figure 4.7).

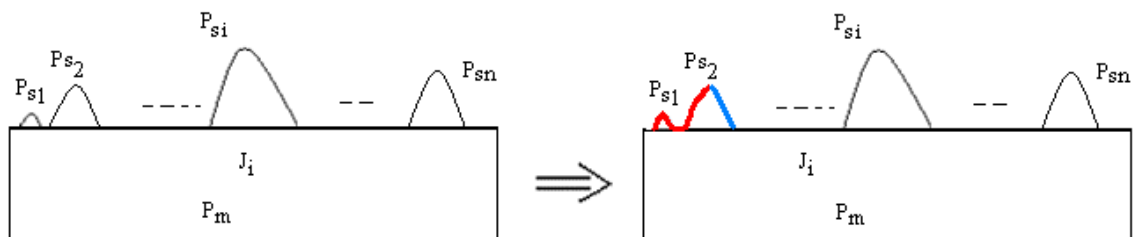


Figure 4.7 Exemple d'élimination d'une partie bruitée située à gauche des parties secondaires

Remarque :

Cette proposition est aussi applicable pour le cas d'une partie composée de la forme :
 $\langle CP \rangle P_m D_i P_{s1} P_{s2} \dots P_{si} \dots P_{sn} \langle /CP \rangle$

Proposition 2 :

Soient :

- $\langle CP \rangle P_{s1} P_{s2} \dots P_{s(i-1)} P_{si} \dots P_{sn} J_i P_m \langle /CP \rangle$ une partie composée du descripteur requête.
- Toutes les parties secondaires (P_{si}) sont de longueur inférieure au seuil $LONG_{SEUIL}$.

Les parties (P_{si}) sont alors considérées issues d'un bruit et seront donc éliminées de l'expression de la partie composée. La nouvelle partie composée devient (voir figure 4.8):

$$\langle CP \rangle P'_m \langle /CP \rangle$$

Les frontières des parties secondaires ainsi que les segments de la ligne de jonction ayant les attributs « h » et « w » seront la continuation des frontières gauche et droite de la partie P_m . Le point début des frontières de P'_m sera le point le plus haut des parties secondaires (voir figure 4.8).

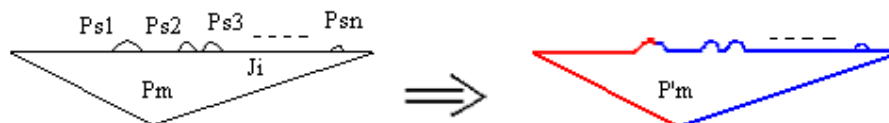


Figure 4.8 Cas des parties secondaires bruitées

L'application de ce processus de réduction est récursive. Pour un seuil $LONG_{SEUIL}$ choisi les propositions 1 et 2 sont appliquées sur les parties composées dont les longueurs des parties secondaires sont inférieures au seuil $LONG_{SEUIL}$. Un nouveau descripteur de la silhouette requête est ainsi obtenu.

Deux cas se présentent :

- Soit l'index du descripteur requête est reconnu dans la table des index, dans ce cas le processus de comparaison continue.
- L'index du descripteur requête n'est pas reconnu dans la table des index, dans ce cas le seuil $LONG_{SEUIL}$ est incrémenté et le processus de transformation du descripteur

requête est de nouveau exécuté. Le fait d'incrémenter la valeur du seuil est synonyme de la considération que la forme a été sujette à un bruit plus important.

L'incrémentation du seuil $LONG_{SEUIL}$ se fera jusqu'à ce que l'index du descripteur requête ait été reconnu, ou le seuil $LONG_{SEUIL}$ dépasse la valeur autorisée $LONG_{SEUIL-MAX}$.

Sans donner la description détaillée des parties, la figure 4.9 illustre trois exemples de réduction du descripteur de la partie composée $\langle CP \rangle P_0 P_1 P_2 J_1 P_3 \langle /CP \rangle$ en un nouveau descripteur comme suit:

- Dans la figure (4.9.a), P_2 est une partie bruitée, la nouvelle partie composée est $\langle CP \rangle P_0 P'1 J'1 P_3 \langle /CP \rangle$, deux couleurs différentes indiquent les parties gauche et droite de la partie $P'1$.
- Dans la figure (4.9.b), P_1 est une partie bruitée, la nouvelle partie composée est $\langle CP \rangle P_0 P'2 J'1 P_3 \langle /CP \rangle$, deux couleurs différentes indiquent les parties gauche et droite de la partie $P'2$.
- Dans la figure (4.9.c), P_0 , P_1 et P_2 sont des parties bruitées, la nouvelle partie composée est $\langle CP \rangle P'3 \langle /CP \rangle$, deux couleurs différentes indiquent les parties gauche et droite de la partie $P'3$. Dans ce cas toutes les parties secondaires sont bruitées.

La figure 4.10 illustre trois autres exemples de réduction du descripteur de la partie composée $\langle CP \rangle P_0 D_1 P_1 P_2 P_3 \langle /CP \rangle$ en un nouveau descripteur comme suit:

- Dans la figure (4.10.a), P_2 est une partie bruitée, la nouvelle partie composée est $\langle CP \rangle P_0 D'1 P'1 P_3 \langle /CP \rangle$, deux couleurs différentes indiquent les frontières gauche et droite de la partie $P'1$.

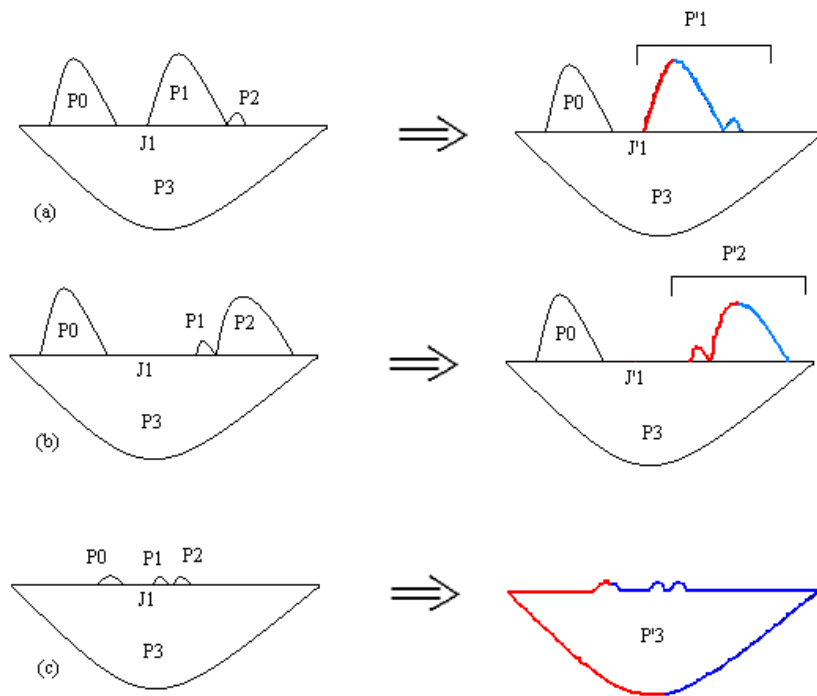


Figure 4.9 Transformation de descripteurs dans le cas d'une ligne de jonction

- Dans la figure (4.10.b), P_1 est une partie bruitée, la nouvelle partie composée est $\langle CP \rangle P_0 D'1 P'2 P_3 \langle /CP \rangle$, deux couleurs différentes indiquent les frontières gauche et droite de la partie $P'2$.
- Dans la figure (4.10.c), P_0 , P_1 et P_2 sont des parties bruitées, la nouvelle partie composée est $\langle CP \rangle P'3 \langle /CP \rangle$, deux couleurs différentes indiquent les frontières gauche et droite de la partie $P'3$. Dans ce cas toutes les parties secondaires sont bruitées.

Les figures 4.9 et 4.10 sont tracées pour mieux expliquer le processus de réduction de descripteurs. Toutes les transformations sont appliquées uniquement sur les descripteurs textuels et non pas sur les images.

Nous donnons ci-après l'algorithme de recherche d'un descripteur requête en opérant un ensemble de transformations pour réduire les effets du bruit.

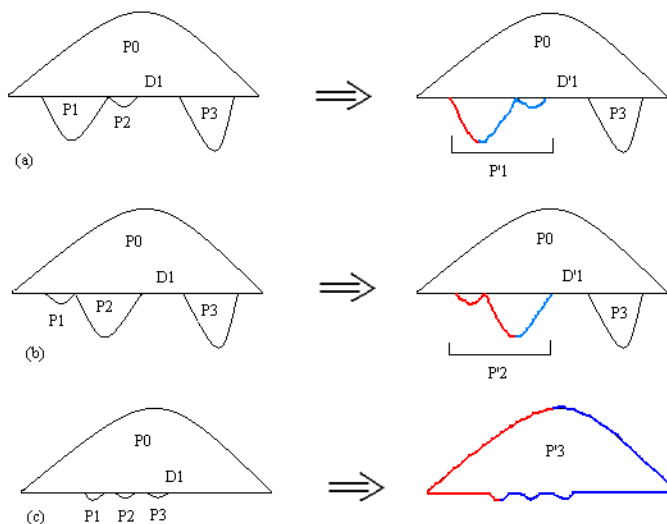


Figure 4.10 Transformation de descripteurs dans le cas d'une ligne de disjonction

Algorithme pour la recherche

L'algorithme suivant est schématisé par l'organigramme de la figure 4.11.

Soient S_R la silhouette requête et I_R l'index de son descripteur textuel. Notre but est de vérifier s'il existe un index similaire dans la base d'index des silhouettes modèles [13].

Algorithme Recherche-Index-Descripteur-1

Début

Soient :

- LONG la taille de la partie considérée.
- $LONG_{SEUIL}$ est la longueur seuil au dessous de laquelle les parties ayant une longueur LONG inférieure à ce seuil sera sujette à l'application de la réduction comme décrit par les propositions 1 et 2.
- $LONG_{SEUIL-MAX}$ un seuil limite pour le seuil $LONG_{SEUIL}$

1. Fixer $LONG_{SEUIL-MAX}$
2. Initialiser $LONG_{SEUIL}$ à zéro
3. Calculer l'index I_R de S_R

Tant que (I_R n'appartient pas à la table des index de la base des descripteurs) et

($LONG_{SEUIL} \leq LONG_{SEUIL-MAX}$)

Faire

- $LONG_{SEUIL} = LONG_{SEUIL} + 1$
- Appliquer le processus de réduction pour toutes les parties dont les tailles LONG sont inférieures à $LONG_{SEUIL}$.
- Calculer l'index du nouveau descripteur.

Fait;

Si ($LONG_{SEUIL} \leq LONG_{SEUIL-MAX}$) **Alors** l'index de la silhouette modèle correspondante est trouvé

FinSi;

Fin.

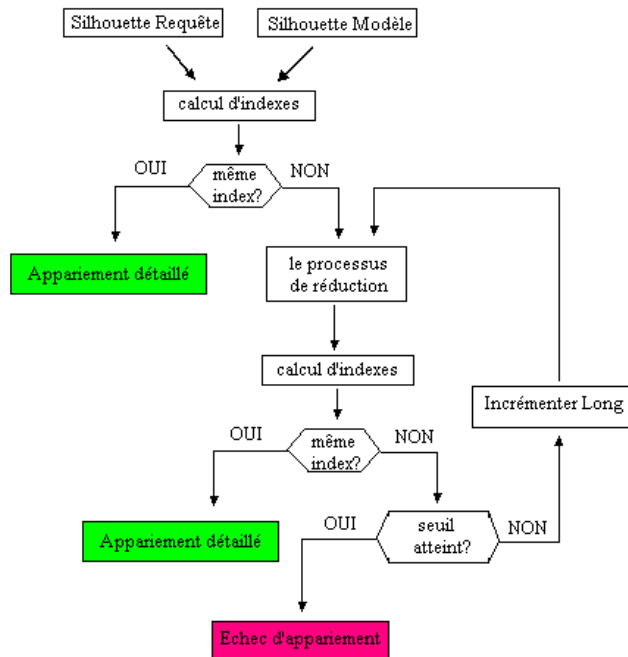


Figure 4.11 Organigramme de recherche de descripteur requête dans une base de descripteurs

Exemple

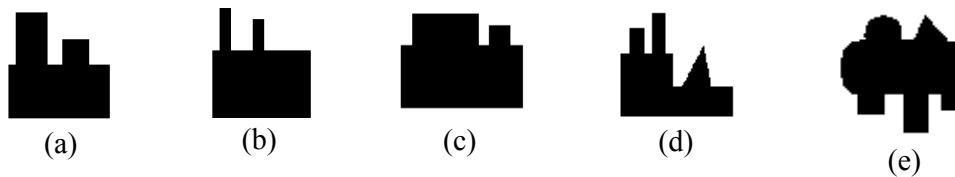


Figure 4.12 Indexation de descripteurs

Une condition nécessaire pour qu'une silhouette requête soit mise en correspondance avec une silhouette modèle est que leurs descriptions XLWDOS aient le même index. Tous les descripteurs ayant le même index que la requête seront sélectionnés pour l'étape suivante.

Supposons que la silhouette (a) de la figure 4.12 est une silhouette requête et que les autres sont des modèles. Les silhouettes modèles (b) et (c) seront candidates pour la reconnaissance car elles ont le même index que la silhouette (a). En effet chacune des trois silhouettes contient deux parties suivies d'une ligne de jonction puis d'une troisième partie. La description de chaque ligne et le nombre de parties qui lui sont attachées est la même. Par contre les silhouettes (d) et (e) ne possèdent pas les mêmes index que la silhouette (a).

4.3.3.3 Méthode de lissage de silhouettes

La seconde méthode que nous proposons pour pallier au problème du bruit altérant la frontière de la silhouette consiste à retrouver l'index du descripteur requête dans la table des index de la base des descripteurs en appliquant un lissage du contour de la silhouette requête qui permet d'éliminer les parties supposées dues au bruit [118]. L'algorithme de lissage est effectué avec le filtre gaussien $G_{\sigma}(x,y)$. Il prend en entrée une silhouette et la valeur du paramètre σ du filtre. Plus la valeur du paramètre σ du filtre Gaussien est grande, plus le contour externe est adouci.

Le lissage par le filtre gaussien est utilisé parce qu'il permet la représentation des silhouettes à différentes échelles ce qui est adaptable pour notre approche [149, 147], c'est à dire éliminer le bruit de 1 pixel puis deux pixels, etc. Le choix de ce filtre est aussi justifié par la possibilité de moduler le paramètre σ qui est nécessaire pour la prise en compte des différents types de déformations.

Nous proposons donc de rechercher l'index du descripteur requête dans la table des index. Dans le cas où il n'appartient pas à cette table, nous proposons d'appliquer un lissage de la frontière de la silhouette requête et de chercher de nouveau l'index de son descripteur. Si celui-ci n'est pas trouvé, un nouveau lissage est opéré mais avec une valeur du paramètre σ plus importante. Ce processus est répété tant que l'index du descripteur calculé n'est pas dans la table des index ou bien si la valeur du paramètre n'a pas dépassé un seuil σ_{MAX} .

Afin de ne pas calculer le masque de convolution à chaque fois que la valeur de σ est incrémentée, nous appliquons le filtre sur le résultat de la convolution [119] suivi par un seuillage par la méthode Isodata pour obtenir la silhouette lissée.

Nous obtenons ainsi une classe de descripteurs requêtes correspondant à différentes échelles. Seul le descripteur dont l'index a été retrouvé dans la table des index sera considéré pour la phase de la comparaison détaillée.

L'algorithme de recherche de l'index du descripteur d'une silhouette est donné par :

Algorithme Recherche_Index_Descripteur-2

Début

{Initialisation}

S est la silhouette initiale.

σ_{MAX} est la valeur maximale autorisée pour le paramètre σ

Initialiser les valeurs de $\sigma = \sigma_0$ et σ_{MAX} .

Calculer le descripteur XLWDOS de S.

Tant que (L'index du descripteur de S n'appartient pas à la table des index) et
($\sigma \leq \sigma_{MAX}$)

Faire

- Incrémenter la valeur de σ
- $S \leftarrow S \otimes G_{\sigma}(x,y)$
- Extraire le contour correspondant à S
- Calculer le nouveau descripteur XLWDOS de S et son index

Fait.

FIN.

La figure 4.13 illustre l'application de la convolution sur une silhouette de synthèse. Plus de détails sur l'utilisation du filtre gaussien est donné en annexe A.

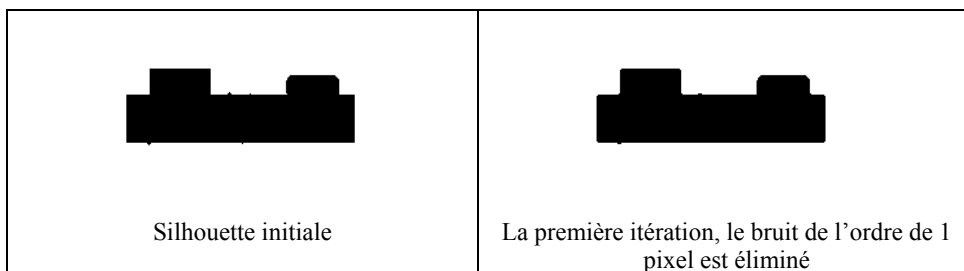


Figure 4.13 Application de l'Algorithme de lissage sur une silhouette.

Montrons sur un exemple l'application de l'algorithme Recherche_Index_Descripteur-2 sur une image d'un objet isolé. (Voir figure 4.14). La silhouette de l'objet requête est extraite de l'image comme le montre la figure 4.15. Malgré que les deux contours extraits sont visiblement similaires, leurs descripteurs sont différents car l'image requête est bruitée (figure 4.15.a)

Le résultat de leurs partitionnement est illustré par les mêmes figures, on remarque que la silhouette requête présente une ligne de jonction et une partie en plus par rapport à la silhouette modèle.

Comme les deux descripteurs calculés ne sont pas identiques, l'image requête est lissée avec une valeur de $\sigma=1,318$. Les deux descripteurs calculés (figure 4.15.b) ne sont toujours pas identiques, la silhouette requête est de nouveau lissée pour $\sigma= 1.86$.

Les deux descripteurs des silhouettes requête et modèle ont donc le même index après le deuxième lissage.

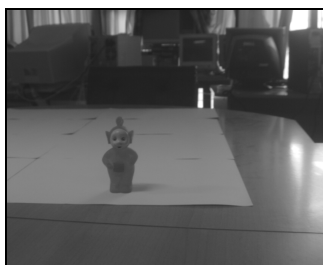


Figure 4.14 image d'un objet

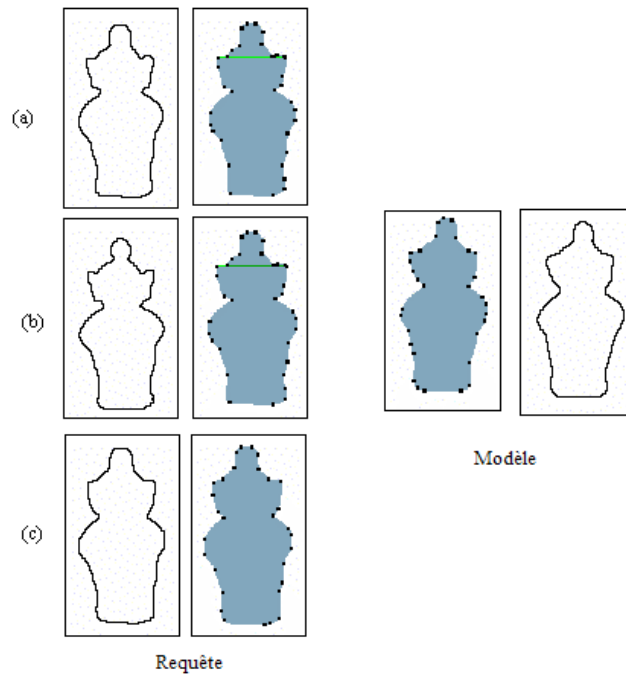


Figure 4.15 Exemple d'application de l'algorithme pour différentes valeurs de σ

4.4. Mesure de similarité entre descripteurs requête et modèle

Ayant un descripteur requête et un ensemble de descripteurs modèles de même index, il s'agit de classer les silhouettes modèles selon leur similarité à la silhouette requête.

Pour ce faire, nous proposons deux mesures de similarité :

La première est globale, elle classe les silhouettes modèles selon le critère des surfaces de leurs parties relativement à la surface des parties de la silhouette requête.

La seconde concerne la géométrie des frontières des parties. Nous proposons une mesure permettant de quantifier la similarité des frontières des parties des silhouettes modèles avec celles de la silhouette requête.

4.4.1 Mesure de similarité : Facteur global

4.4.1.1 Principe de base

Les silhouettes concernées par la mesure de similarité sont celles obtenues à l'étape précédente, elles ont donc toutes la même structure, et le même index que la requête et en particulier, le même descripteur XLWDOS global que celui de la requête.

La mesure de similarité des descripteurs globaux correspond au calcul de la surface de chaque partie et moyennant leur description textuelle.

Le premier facteur de similarité (SF_1) entre les silhouettes mesure la différence entre les surfaces des parties en correspondance. Il est donné par l'équation suivante :

$$SF_1 = \sum_{i=1}^{NbParts} (S_{mi} - S_{ri})^2$$

où :

- NbParts désigne le nombre de parties de chaque silhouette
- S_{mi} , S_{ri} sont les surfaces des parties numéro i des silhouettes modèle et requête

Pour ce faire, il est nécessaire de réaliser les tâches suivantes :

Reconstruction des silhouettes à partir de leurs descripteurs

Il est important de rappeler que dans notre approche nous nous basons uniquement sur la description textuelle des silhouettes par conséquent les coordonnées des points utilisés dans nos calculs sont déduites de cette description.

Ramener les deux silhouettes dans la même échelle

Il s'agit de représenter les coordonnées des points reconstruits des deux silhouettes dans un même repère attaché au rectangle d'aire minimum englobant une des silhouettes en appliquant une homothétie.

Calcul des différences des surfaces des parties en correspondance

Une partie de la silhouette requête ou modèle est composée d'un ensemble de contours élémentaires (courbe convexe, courbe concave, segment de droite) où chacun est délimité par deux extrémités (début et fin). Chaque contour courbé sera approximé par un ensemble de segments.

La surface d'une partie est constituée de deux frontières : gauche et droite (voir figure 4.16). Pour calculer sa surface, nous utiliserons la formule des trapèzes (voir la figure 4.17).

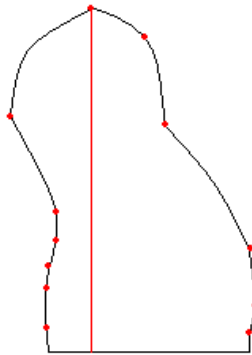


Figure 4.16 Exemple d'une partie d'une silhouette

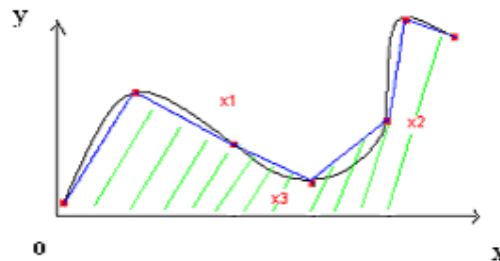


Figure 4.17 Principe de l'application de la méthode de trapèzes

Soit $f(x)$ la fonction décrivant la courbe. a et b sont respectivement le début et la fin de l'intervalle. $f(a)$ et $f(b)$ sont les valeurs de la fonction en ces points (voir figure 4.19).

Le calcul de la surface de la courbe se fait selon la formule des trapèzes en discrétisant l'intégrale en une somme de valeurs de la courbe.

$$\int_a^b f(x) dx \approx (b-a)/2 * [f(a) + f(b)]$$

En effet considérons la figure 4.18 et calculons l'aire hachurée.

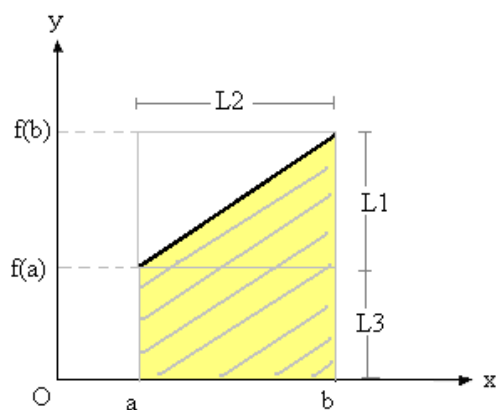


Figure 4.18 Exemple d'application de la méthode de trapèzes

Les longueurs L1, L2 et L3 correspondent aux expressions suivantes (voir figure 4.18)

$$L1 = f(b) - f(a), \quad L2 = b - a \quad \text{et} \quad L3 = f(a)$$

La surface hachurée S est :

$$S = (L1 * L2) / 2 + L3 * L2 = L2 / 2 (L1 + 2 * L3)$$

En remplaçant L1, L2 et L3 par leurs expressions correspondantes nous trouvons :

$$S = (b - a) / 2 * [f(a) + f(b)].$$

Les détails concernant le calcul de la surface d'une partie est donné en annexe B.

Considérons les deux courbes f et g de la figure 4.19. Pour que ces deux courbes tendent l'une vers l'autre il faudrait qu'elles génèrent la même surface relativement à l'axe (ox). Autrement dit la différence de surfaces générées doit tendre vers zéro [10]. Comme les

ordonnées et les abscisses des points sont toutes positives grâce à l'utilisation du rectangle minimum comme repère, cela se traduit par :

$$\left| \int_a^b f(x) dx - \int_a^b g(x) dx \right| \sim 0$$

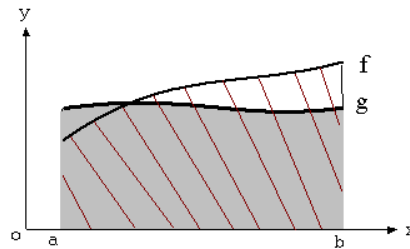


Figure 4.19 La différence de l'aire entre deux courbes

C'est sur ce principe que se base le calcul de la différence de surfaces des silhouettes à partir de leurs descripteurs.

La seconde étape consiste à calculer les vecteurs $V_r(S_{r1}, S_{r2}, \dots, S_{rn})$ et $V_m(S_{m1}, S_{m2}, \dots, S_{mn})$ des surfaces pour chacune des silhouettes requête et modèle.

Une fois les différences de surfaces calculées, le meilleur modèle sera donc celui pour lequel cette différence est minimale. Cette mesure tend vers zéro dans le cas où la ressemblance est très importante entre les silhouettes à comparer.

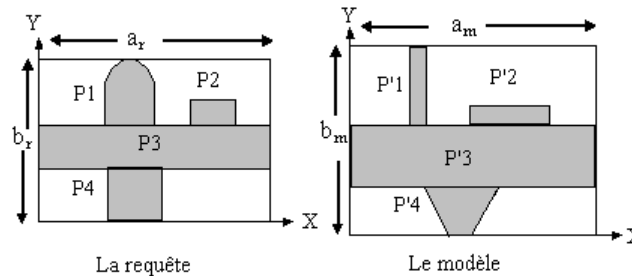


Figure 4.20 Mesure de similarité entre silhouettes

Considérons l'exemple de la figure 4.20, où l'image modèle et requête ont toutes les deux le même index. Les dimensions du rectangle minimum de la requête sont données par les valeurs a_r , b_r , celles du modèle sont données par a_m , b_m . Rappelons que ce rectangle est

unique [120, 80] c'est pourquoi nous le prenons comme repère de travail conformément à la démarche de XLWDOS. Pour effectuer la comparaison nous ramenons d'abord les coordonnées (X_i, Y_i) des points de la requête à (X'_i, Y'_i) relativement au repère du modèle et ce en appliquant une homothétie.

Nous obtenons ainsi : $X'_i = (a_m/a_r) X_i$ et $Y'_i = (b_m/b_r) Y_i$.

Pour calculer la mesure de similarité, nous calculons les surfaces occupées par chaque partie de la silhouette. Notons par $S(P_i)$ la surface de la partie P_i , la valeur de similarité calculée pour les silhouettes de la figure 4.20 sera: $[S(P1)-S(P'1)]^2+[S(P2)-S(P'2)]^2+[S(P3)-S(P'3)]^2+[S(P4)-S(P'4)]^2$

Le calcul de la surface d'une partie est fait en appliquant la formule des trapèzes qui approxime une courbe par une succession de segments. Dans le cas d'une partie ayant des portions de courbe (comme $P1$ de la figure 4.20), les seuls points fournis par la description sont les points extrémités ainsi que le point de la courbe le plus éloigné du segment joignant les points extrémités. Un troisième point de la courbe est généré à partir du descripteur situé à une distance maximale du segment reliant ses deux extrémités (voir figure 4.21).

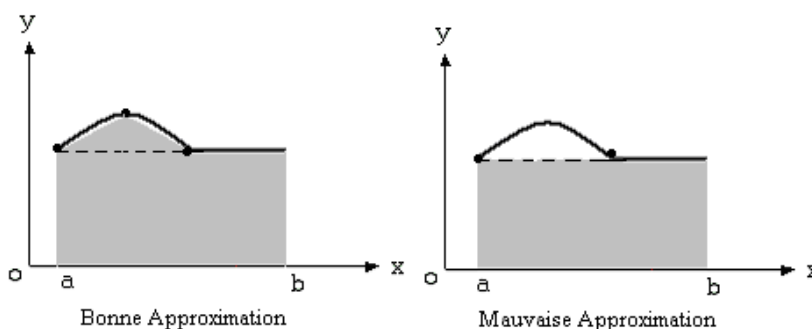


Figure 4.21 Approximation de surfaces par la méthode des trapèzes

Une fois la mesure de similarité calculée, le meilleur modèle sera donc celui pour lequel cette différence est minimale. Cette mesure tend vers zéro dans le cas où la ressemblance est très importante entre les silhouettes à comparer.

Cette mesure constitue une condition nécessaire mais pas suffisante pour la reconnaissance, c'est pourquoi nous avons besoin d'un facteur local pour la comparaison.

4.4.2 Mesure de similarité : facteur local

4.4.2.1 Principe de base

Les silhouettes modèles résultant de l'étape de l'indexation seront candidates pour la reconnaissance. Le recours aux différences de surfaces est un critère qui permet de faire une première sélection des silhouettes similaires.

Etant donné que l'utilisation des surfaces n'est pas une condition suffisante pour sélectionner le bon modèle, on a recours à l'étude de la géométrie de chaque élément.

Pour que les descripteurs détaillés de deux silhouettes (requête et modèle) soient comparables, il faudrait que la géométrie des éléments qu'ils décrivent soit quasiment la même. On entend par élément tous les attributs XLWDOS décrivant chaque partie et ligne de séparation de la silhouette.

Pour ce faire nous utiliserons la méthode des quasi-invariants, dont le principe de base est que le mouvement apparent entre deux images successives peut être approximé par une similitude [114, 86]. En effet l'angle et le rapport de distance peuvent être considérés des invariants géométriques pour deux images dont les prises de vues sont très proches, d'où leur noms de quasi-invariants.

Supposons que la silhouette de la figure 4.22 corresponde à une silhouette requête.

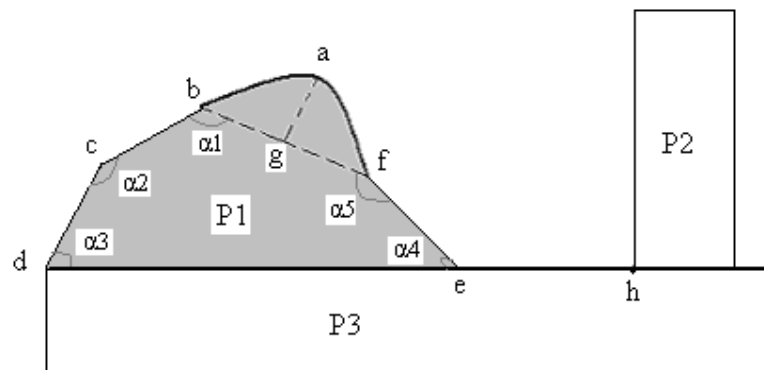


Figure 4.22 Comparaison des descripteurs détaillés

Considérons à titre d'exemple α_1 l'angle entre les segments (fb) et (bc) et L_1 le rapport de distance (fb/bc). Une image modèle correspondant à cette requête doit avoir un angle α_1 et une longueur L_1 quasiment identiques à ceux de la requête. Autrement dit, plus les deux formes sont similaires, plus les différences de leurs angles et des différences des rapports des distances de leurs segments tendent vers zéro.

Notre but consistera donc à trouver cet intervalle pour lequel on dira que deux angles ou deux rapports sont quasiment les mêmes.

Pour mener à bien cette étude il est important de travailler sur des centaines d'images fournissant des dizaines de milliers de descripteurs afin de déterminer d'une manière efficace les variations des quasi-invariants géométriques.

4.4.2.2 Etude de variation des quasi invariants

L'hypothèse de base des quasi invariants est que le déplacement existant entre deux images représentant le même modèle peut être approché par une similitude, ceci suppose que le changement de point de vue soit modéré. Ce qui permet de caractériser les images par des quasi-invariants comme ceux utilisés par GROS et LAMIROY [114, 86] et qui sont l'angle formée par deux segments adjacents et leur rapport de longueurs.

L'utilisation des quasi-invariants vient remplacer celle des invariants projectifs qui furent utilisés auparavant pour la reconnaissance [116], et ce, malgré qu'il n'existe pas d'invariants formels lors d'une transformation projective de R^3 vers R^2 [114]. Mais leur fragilité par rapport au bruit et leur pouvoir descriptif un peu faible [16] les rend moins favorisés que les quasi-invariants pour leur utilisation à des fins de reconnaissance.

A titre d'exemple, les figures 4.23 et 4.24 illustrent le classement des modèles appartenant à une base de modèles selon leur ressemblance à l'image requête indiquée (image d'une disquette). Il s'agit en fait des meilleurs modèles sélectionnés par la reconnaissance en utilisant la méthode des quasi-invariants. La reconnaissance sélectionne le bon objet pour la requête (vote=63) tandis que le modèle qui vient juste après a une valeur de vote relativement faible.

Notre méthode de reconnaissance en utilisant les quasi invariants [11] est basée sur la technique d'indexation géométrique étendue [114], en ajoutant des modifications au niveau du stockage des données et du vote. Ces modifications consistent en l'utilisation d'un quadtree [24, 45] pour l'indexation des descripteurs afin d'optimiser la phase de recherche, ainsi que la technique du VA-File pour l'étape de vote [206].



Figure 4.23 Exemple de requête pour la reconnaissance avec les quasi-invariants

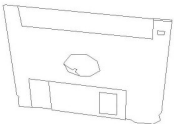
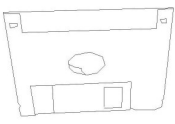
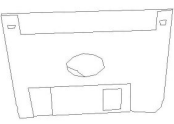
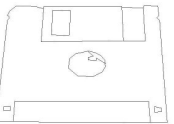


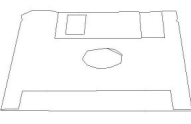
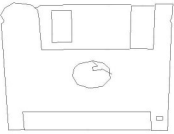
			
Image 1 (Vote: 63)	Image 2 (Vote: 12)	Image 3 (Vote: 12)	Image 4 (Vote: 11)
			
Image 5 (Vote: 10)	Image 6 (Vote: 9)	Image 7 (Vote: 9)	Image 8 (Vote: 8)

Figure 4.24 Les meilleurs modèles sélectionnés pour la requête

Deux quasi-invariants seront utilisés, l'angle θ formé par deux segment adjacents, et leur rapport de longueurs ρ (Voir figure 4.25).

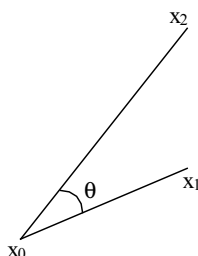


Figure 4.25 Les quasi invariants géométriques

$$\rho = \frac{\overrightarrow{x_0x_1}}{\overrightarrow{x_0x_2}} \quad \theta = \arccos \frac{\overrightarrow{x_0x_1} \cdot \overrightarrow{x_0x_2}}{\|\overrightarrow{x_0x_1}\| \|\overrightarrow{x_0x_2}\|}$$

Pour éviter tout problème de confusion, par convention l'ordre des segments est choisi tel que l'angle θ soit toujours positif et minimum.

Pour chaque configuration de deux segments, les deux valeurs θ et ρ extraites seront toujours les mêmes quelque soit la position et la direction de cette configuration. Ceci permettra de déterminer correctement la similitude reliant deux configurations similaires sur deux images voisines.

Puisque une configuration est formée de trois points de R2 linéairement indépendants, nous aurons besoin de six attributs (figure 4.26) :

- L'angle θ formé par les deux segments.
- L'angle θ_0 formée par le premier segment et la semi-droite [OX.
- Les longueurs des deux segments : l_1 et l_2 .
- Les coordonnées du point d'intersection des deux segments x_0, y_0 .

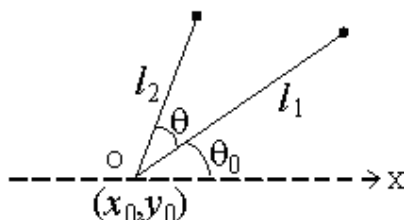


Figure 4.26 Les éléments formant un descripteur

La structure de ces descripteurs étant formée, ils seront indexés suivant les valeurs de l'angle θ et du rapport $\rho = l_1/l_2$. Le choix de ces deux valeurs comme clé de stockage vient du fait qu'elles sont invariantes aux similitudes.

L'hypothèse de base prévoit l'existence d'une similitude entre les images requête et modèle, si elles représentent le même objet. Une condition nécessaire de la validité de la mise

en correspondance entre deux configurations est que leurs angles et rapports de distance soient relativement proches.

Mais plutôt que de calculer ρ à chaque fois à partir de l_1 et l_2 , il sera plus judicieux de l'intégrer dans la structure du descripteur pour optimiser le temps de recherche.

Les évaluations ont porté sur 856 images représentant 28 objets. Ces images ont été générées par un logiciel d'animation 3D RayDream®. Une des caractéristiques importantes des images générées par ce logiciel (voir figure 4.27) est qu'elles ont une apparence quasi réelle (irrégularité des surfaces).

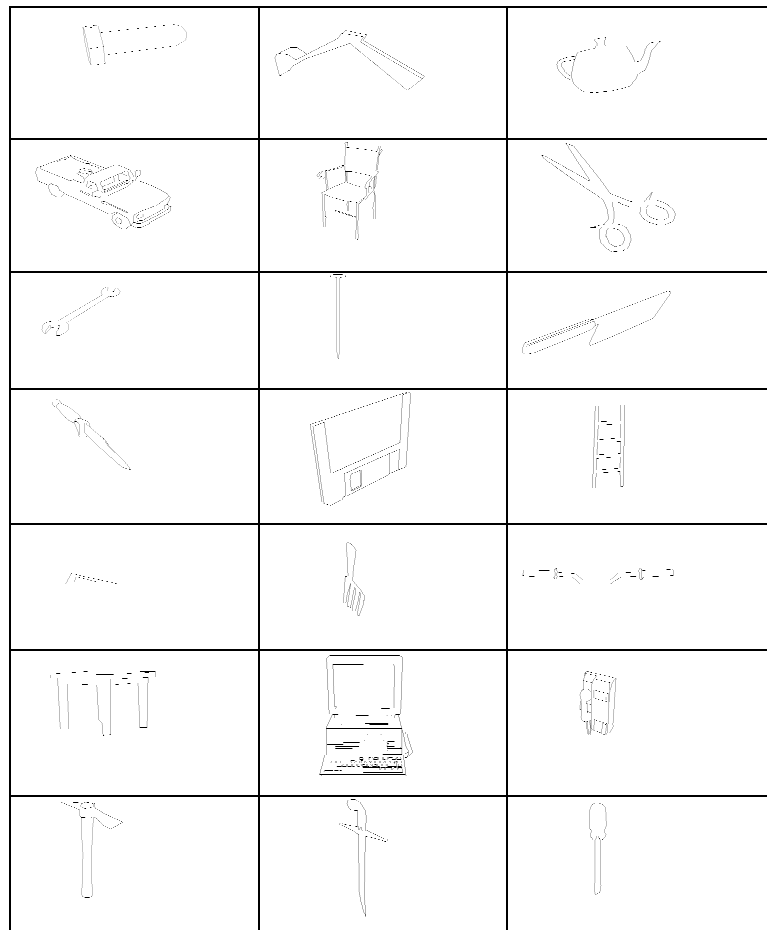


Figure 4.27. Les objets utilisés pour l'évaluation

Pour chaque objet, des vues sont prises à des intervalles de 20° sur deux axes de rotation différents. Les vues sont ensuite passées dans un logiciel de traitement d'images afin d'uniformiser les couleurs des surfaces (changer les textures en couleurs) puis extrait les contours et les approche par des segments de droites. Le résultat de ce traitement de bas niveau est un ensemble de coordonnées des segments formant l'image d'origine (figure 4.28).

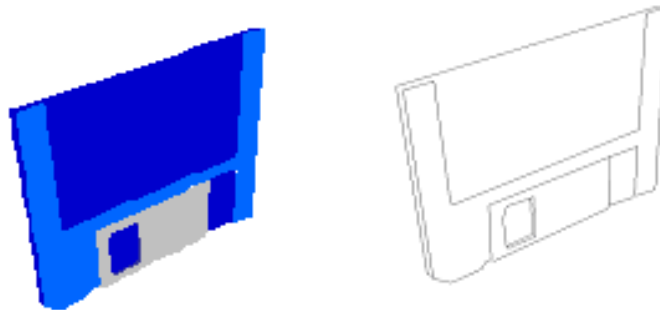


Figure 4.28 Exemple d'une image générée par RayDream et le résultat de sa segmentation

4.4.2.2.1 Étude Statistique sur les éléments de l'index (θ , ρ)

Afin de bien mener l'implémentation de la phase d'indexation des descripteurs, nous avons réalisé une étude statistique sur la distribution des éléments de l'index (l'angle θ et le rapport ρ). Cette étude a porté sur 49 764 descripteurs extraits des 856 images.

a) L'angle (θ) formé par les deux segments

Nous avons commencé par étudier la distribution des descripteurs suivant l'angle θ , et nous avons obtenu le graphe ci-dessous (figure 4.29).

Nous remarquons un pic qui se situe autour de $\pi/2$, un angle qui apparaît souvent dans les configurations. Aussi, il y a une concentration se situant autour des grands angles, ce qui est expliqué par la mauvaise qualité de la segmentation qui, pour certaines configurations, transforme un angle assez petit, en deux angles dont l'un est d'une grande valeur (exemple dans la Figure 4.30).

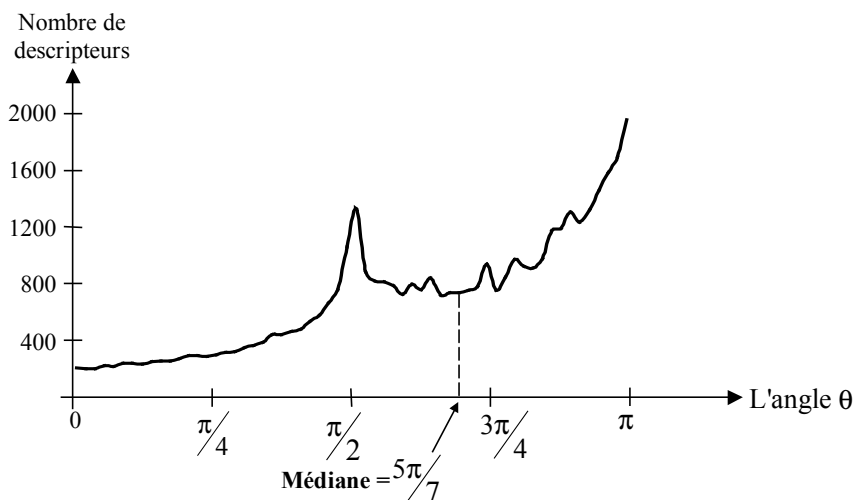


Figure 4.29 Distribution des descripteurs suivant l'angle θ



Figure 4.30 Exemple d'une transformation engendrée par la segmentation

Le rapport (ρ) des longueurs des deux segments

On veut calculer le rapport $\rho = 11/12$ de deux segments. Théoriquement, la probabilité pour que le premier segment soit plus long que le second segment est égale à celle où il est moins long. Donc, le nombre de configurations dont le rapport $\rho \geq 1$ ($\rho \in [1, +\infty[$) est égal à celui des configurations dont le rapport $\rho \leq 1$ ($\rho \in [0, 1]$) ; d'où une densité très importante dans la distribution des descripteurs dans cette région de l'espace de données (voir la courbe de la figure 4.31)

Dans cette courbe apparaît clairement la forte densité dans l'intervalle $]0,1]$, la médiane étant égale à $0,95$. Les deux intervalles : $]0, 0,95]$ et $[0,95, +\infty[$ contiennent le même nombre de descripteurs.

Donc, pour remédier à ce problème, il suffit de remplacer le rapport ρ par son logarithme népérien $\ln(\rho)$ (figure 4.32) qui assure une répartition uniforme.

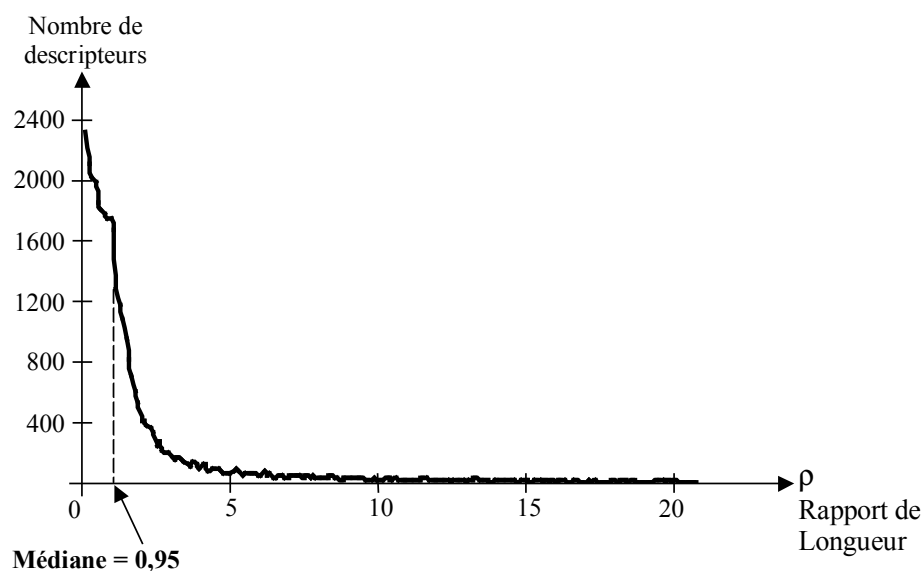
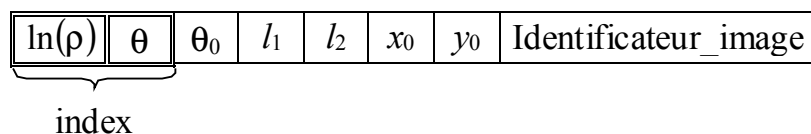


Figure 4.31 Distribution des descripteurs suivant le rapport de longueur ρ

Nous avons exclu les descripteurs dont la valeur absolue du logarithme du rapport dépasse 4, puisque plus de 99% des descripteurs sont contenus dans l'intervalle $[-4,+4]$; et en dépassant cette valeur, le quasi-invariant n'aura pratiquement aucune signification à cause du bruit.

Donc, la forme d'un descripteur sera comme suit (nous avons maintenu $\ln(\rho)$ au lieu de ρ)



4.4.2.2.2 Les seuils de comparaison pour la recherche

Lors de l'opération de recherche, l'algorithme va déterminer parmi les descripteurs de la base, ceux qui se trouvent à un certain rayon du descripteur requête. La zone de recherche est caractérisée par deux paramètres: un seuil de recherche pour l'angle de la configuration (θ), et le deuxième pour le logarithme du rapport de longueur ($\ln(\rho)$).

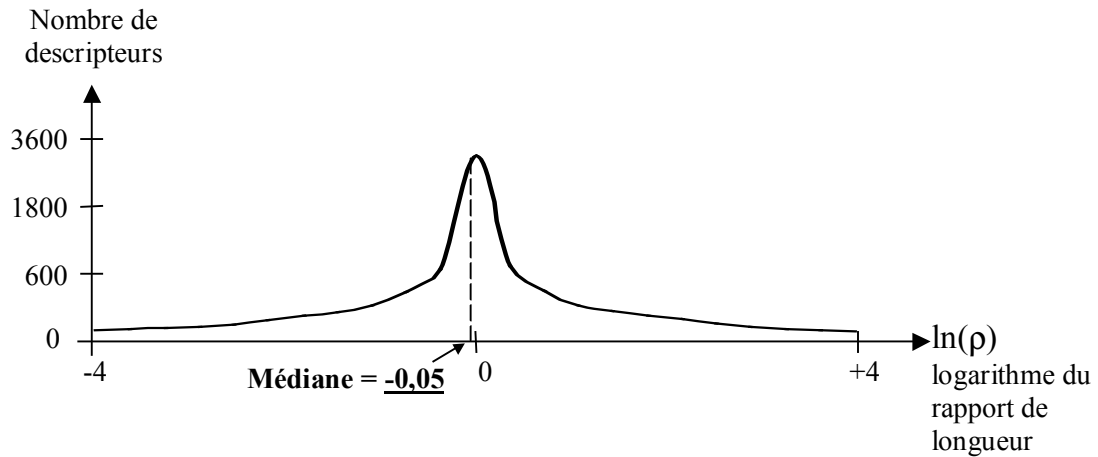


Figure 4.32 Distribution des descripteurs suivant le logarithme népérien du rapport de longueur

Pour déterminer ces deux paramètres, nous avons calculé la différence entre les deux quasi-invariants des mêmes configurations appartenant à des couples d'images successives prises du même objet (voir figure 4.33).

Nous avons pris 24 couples d'images successives; ensuite, pour chaque couple, nous avons pris les couples de configurations identiques et nous avons calculé la différence entre les deux quasi-invariants (les résultats sont montrés dans les deux graphes ci-dessus). La distance entre les quasi-invariants de 90% des couples de configurations est inférieure à ($\ln(\rho)=0.23$, $\theta=18.61^\circ$). Étant donné qu'une image requête va se situer entre deux images successives de la base, donc le seuil de recherche peut être pris comme étant la moitié de la distance entre deux images. Nous avons donné pour le seuil de recherche la valeur de 60% de la distance entre deux images [11].

Donc: Le seuil de recherche pour $\ln(\rho)$ est :

$$\ln(\rho) = 0,23 \times 0,6 \approx 0,14$$

Le seuil de recherche pour θ est :

$$\theta = 18,61^\circ \times 0,6 \approx 11,2^\circ$$

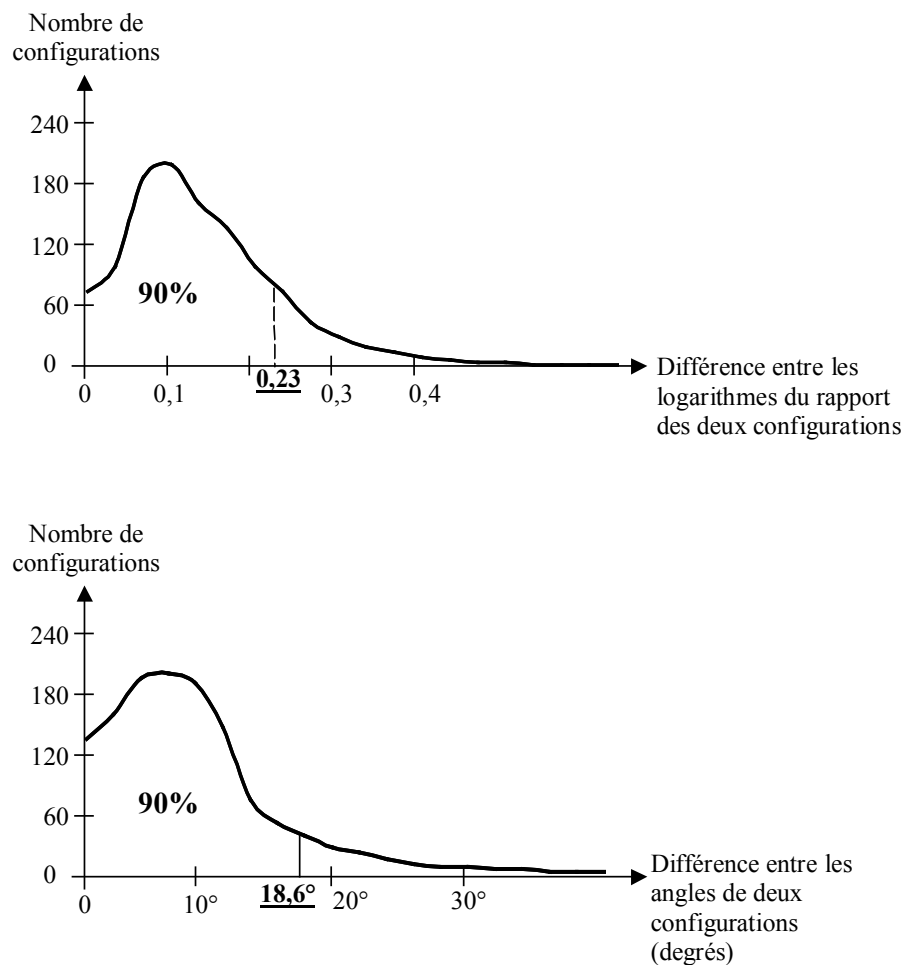


Figure 4.33 Différence entre les quasi-invariants de même configurations appartenant à des couples d'images successives

4.4.2.3 Application des quasi invariants sur les descripteurs détaillés

XLWDOS

Nous utilisons le résultat de cette étude menée sur les quasi-invariants sur les contours élémentaires des silhouettes modèles et requête. Les résultats de la validation de cette mesure seront donnés dans le prochain chapitre.

Pour l'application des valeurs aux descripteurs détaillés XLWDOS, et vu les approximations des descripteurs de silhouettes, nous gardons les seuils initiaux ($\ln(\rho)=0.23$ et $\theta=18.61^\circ$) au lieu des seuils utilisés pour la reconnaissance avec les quasi invariants ($\ln(\rho)=0.14$ et $\theta=11.2^\circ$). Ce choix est justifié par plusieurs raisons :

- L'image requête n'appartient pas à la base des modèles.
- Les descripteurs concernés par la comparaison détaillés ont la même structure, le même index que le descripteur requête, et la différence de leurs surfaces tend vers zéro.
- Le lissage pour une valeur de $LONG_{SEUIL}$ modifie légèrement le contour, donc il sera judicieux d'être aussi tolérant que possible quant aux seuils choisis.
- Le résultat d'application des quasi-invariants est exclusif. Il suffit qu'une seule configuration de l'angle ou de rapport de distance entre la requête et le modèle (même parmi des centaines d'autres) dépasse le seuil pour que le modèle soit éliminé et ne sera plus parmi les candidats pour la reconnaissance.
- Le risque d'avoir plusieurs modèles à la suite de l'application des seuils de comparaison est évité en revenant à leur classement selon les différences des surfaces ce qui nous permettra de sélectionner le bon modèle. Ceci dit le fait de laisser plusieurs modèles selon leur ordre de mérite n'est pas faux et plusieurs travaux de reconnaissance procèdent ainsi.

Exemple

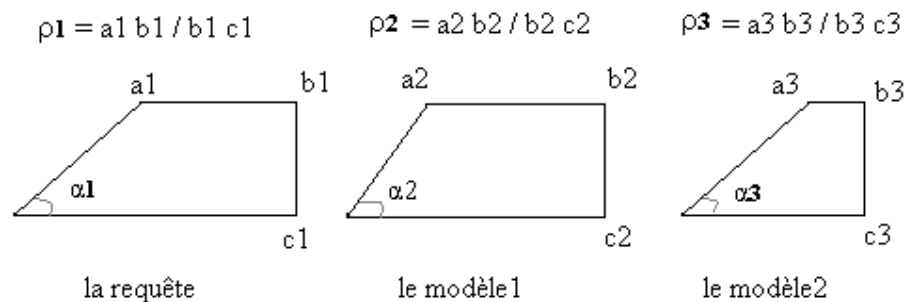


Figure 4.34 Exemple d'application des seuils des quasi-invariants

La figure 4.34 illustre un exemple de parties des silhouettes. A titre d'exemple nous nous intéresserons aux angles α_1 , α_2 et α_3 et aux logarithmes des rapports de longueurs $\ln(\rho_1)$, $\ln(\rho_2)$ et $\ln(\rho_3)$. Même si l'explication porte uniquement sur ces éléments, tous les autres angles et rapport de distances extraites des silhouettes de la figure 4.34 doivent vérifier les

critères des seuils des quasi-invariants. Nous avons déterminé les valeurs suivantes pour les éléments indiqués

$$\alpha_1=45^\circ, \alpha_2=56^\circ \text{ et } \alpha_3=45^\circ.$$

$$\rho_1=1,56 \rightarrow \ln(\rho_1)=0.45.$$

$$\rho_2=1,88 \rightarrow \ln(\rho_2)=0.62.$$

$$\rho_3=0,25 \rightarrow \ln(\rho_3)=-1,39.$$

Comparons la requête avec le modèle1 :

$$|\alpha_1 - \alpha_2| = 11^\circ < \mathbf{18.61^\circ}$$

$$|\ln(\rho_1) - \ln(\rho_2)| = 0.17 < \mathbf{0.23}$$

En supposant que tous les autres éléments (les autres angles et rapports de distances) du modèle1 vérifient les seuils, alors le modèle1 est sélectionné comme celui correspondant à la requête.

Comparons maintenant la requête avec le modèle2 :

$$|\alpha_1 - \alpha_3| = 0^\circ < \mathbf{18.61^\circ}$$

$$|\ln(\rho_1) - \ln(\rho_3)| = 1.84 > \mathbf{0.23}$$

Nous constatons que même si α_1 est égal à α_3 , le modèle2 n'est pas sélectionné car la différence des logarithmes des rapports de distance dépasse le seuil.

4.4 Etude de la complexité algorithmique

4.4.1 Rappel sur la complexité d'un algorithme

Lorsqu'on considère un problème, il peut être intéressant de trouver plusieurs algorithmes pour le résoudre, afin d'utiliser le meilleur d'entre eux. Pour cela trois approches sont possibles :

- L'approche théorique consiste à déterminer mathématiquement la quantité de ressources (temps, espace, etc...)

- L'approche empirique consiste à programmer ces algorithmes et les essayer à l'aide d'un ordinateur.
- L'approche hybride détermine théoriquement le type de fonction qui décrit l'efficacité, puis évalue empiriquement les paramètres numériques dépendant d'une implémentation particulière.

L'approche théorique a l'avantage :

- de ne dépendre ni de l'ordinateur, ni du langage de programmation.
- d'éviter l'effort de programmer inutilement un algorithme inefficace et de gaspiller ensuite le temps de l'ordinateur qui l'essayerait.
- de connaître l'efficacité des algorithmes étudiés quelle que soit la taille des exemplaires sur lesquels ils auront à travailler, alors qu'on aurait peut être tendance avec l'approche empirique, à ne comparer les algorithmes que sur des exemplaires de taille relativement modeste.

On distingue trois sortes de complexités :

- La complexité dans le pire des cas : calcul du coût dans le pire des cas.
- La complexité en moyenne : on calcule le coût pour chaque donnée possible puis on divise la somme de ces coûts par le nombre de données différentes.
- La complexité dans le meilleur des cas : on calcule le coût en se plaçant dans le meilleur des cas.

Nous nous intéressons *au pire des cas* parce que la plupart des algorithmes ont des performances identiques dans le meilleur des cas. Par exemple, trier une liste de deux éléments se fera aussi vite avec n'importe quel algorithme de tri. Au contraire, le pire des cas permet de distinguer les algorithmes les plus performants.

Le temps d'exécution du programme dépend du compilateur et de la machine utilisée pour compiler et exécuter le programme, c'est pourquoi on n'exprime pas le temps d'exécution en unités standard, comme la seconde. Il est préférable alors de dire par exemple que « la complexité de tel algorithme est proportionnelle à n^2 ». La constante de

proportionnalité n'est pas précisée dans ce type de formulation, car elle est presque entièrement déterminée par les performances du compilateur et de la machine, ainsi que par d'autres facteurs comme les entrées/sorties du système.

La notation « grand O » est la plus fréquemment utilisée pour évaluer les performances d'un algorithme. En fait, pour chaque algorithme, l'idéal serait de déterminer une fonction prenant en paramètres le volume des données à traiter, ainsi que les autres paramètres influant sur les performances de l'algorithme s'il y'en a ; mais la rédaction d'une telle fonction peut se révéler très fastidieuse. Utiliser la notation O permet rapidement d'avoir une idée globale des performances de l'algorithme et elle se rapproche souvent des résultats obtenus avec une fonction plus précise. Ce qui nous intéresse lors de l'évaluation d'un algorithme est la rapidité avec laquelle ses performances se dégradent quand le volume de données à traiter augmente.

Avec une fonction, on représente la courbe de croissance de l'algorithme, qui décrit son efficacité. La notation O permet de décrire une courbe proche de celle-ci, et ce sans avoir à rédiger de fonction. On appelle cette courbe Ordre de grandeur (d'où le « O » de la notation).

4.4.2 Complexité de l'algorithme de reconnaissance

Comme nous l'avons déjà vu, notre algorithme de reconnaissance effectue les traitements que nous rappelons ci dessous et pour lesquels nous calculons la complexité au fur et à mesure:

Algorithme Recherche-Index-Descripteur-1

Pour l'algorithme Recherche-Index-Descripteur-1, l'index du descripteur requête est recherché dans la table des index de taille n. Ce processus (calcul d'index et comparaison avec les index des descripteurs modèles) est répété au pire des cas m fois où m est le nombre de seuils utilisés. Le nombre m est généralement une valeur allant de 1 à 3. La complexité de cet algorithme est donc $O(n)$.

Algorithme Recherche-Index-Descripteur-2

La complexité de cet algorithme est identique au précédent. La seule différence pour ces deux algorithmes réside dans le processus effectué avant d'effectuer la recherche dans la table des index des descripteurs modèles. La complexité est donc $O(n)$.

Mesure de similarité : Facteur global

Le traitement consiste à parcourir les descripteurs globaux d'une requête et d'un modèle simultanément puis calculer les surfaces de chaque couple de parties ce qui donnera une complexité $O(n)$.

Mesure de similarité : Facteur local

De même que pour l'étape précédente, il s'agira de parcourir simultanément un descripteur requête et un descripteur modèle et comparer la différence des angles et de rapport de distances d'où une complexité $O(n)$.

4.5 Conclusion

Nous avons présenté dans ce chapitre, une méthode de reconnaissance d'images en nous basant sur les descripteurs XLWDOS des silhouettes d'objets.

Nous avons montré que ce descripteur est exploitable pour le problème de la reconnaissance d'objets à partir de leurs silhouettes.

Tout d'abord nous avons proposé un schéma pour une base de descripteurs textuels et leur indexation. En second lieu, nous avons proposé une méthode pour la recherche de descripteurs modèles candidats pour une similarité avec un descripteur requête. Pour pallier au problème du bruit altérant les frontières de silhouettes requêtes, nous avons proposé deux méthodes : la première consiste à transformer le descripteur sans modifier sa silhouette, la seconde consiste à opérer un lissage incrémental de sa frontière jusqu'à ce que son index soit trouvé dans la table des index.

En dernier lieu, nous avons proposé une mesure de similarité pour le classement des descripteurs candidats par ordre décroissant de similarité. Une similarité globale basée sur le calcul de surfaces et ensuite une similarité locale basée sur les quasi-invariants sont proposées.

Dans le chapitre suivant, nous présentons les résultats de la confrontation des algorithmes proposés sur des données réelles.

Chapitre 5

Validation de l'algorithme de reconnaissance

5.1 Introduction

Nous présentons et discutons dans ce chapitre les résultats d'expérimentation des algorithmes développés. En plus de quelques images de synthèse, la plupart des images utilisées sont des images réelles prises dans notre laboratoire, ainsi que d'autres images réelles utilisées par d'autres travaux existants et qui se basent sur les silhouettes ; il s'agit de la base d'images de B. Leibe et B. Schiele [125] où chaque objet est représenté par 41 vues différentes.

Les images prises dans notre laboratoire sont des images stéréoscopiques ou des séquences d'images d'objets en mouvement. Ces images sont soumises à un logiciel de traitement d'image ISODATA pour le seuillage après l'extraction de silhouettes puis leurs descripteurs XLWDOS sont générés.

Nous validerons, dans ce chapitre, chaque étape de l'algorithme de reconnaissance :

- Nous commencerons par l'algorithme d'indexation en montrant son applicabilité pour la mise en correspondance de deux images, ainsi que la recherche d'images dans une base indexée. Le but de cette étape est de réduire le nombre de modèles qui resteront pour la suite de l'algorithme de reconnaissance.

- La seconde étape concerne la réduction du descripteur requête afin que son index soit retrouvé dans la table des index de la base des descripteurs modèles.
- Une fois l'index de l'image requête est retrouvé dans la table des index des descripteurs modèles, une mesure de similarité des descripteurs globaux permettra de classer les modèles qui ressemblent le plus à une requête. Le traitement se basera, comme nous l'avons vu, sur le calcul des surfaces des silhouettes.
- La dernière étape se base sur l'application des quasi-invariants géométriques pour sélectionner le meilleur modèle. La description détaillée de ce modèle est quasiment la même que celle de la requête en question.

5.2 Validation de l'algorithme d'indexation

5.2.1 Indexation d'une base d'images modèles

La première application de notre algorithme est l'indexation d'une base de silhouettes modèles. Nous utilisons quelques images construites par B. Leibe et B. Schiele [125]. Chaque objet est représenté par 41 vues différentes.

La figure 5.1 illustre une séquence de formes correspondant aux images d'un chien prises en tournant la caméra avec un angle de 24° .

Dans cette étape d'indexation, l'index de chacune des formes est calculé. Un index fait référence donc à toutes les formes de la base dont les descripteurs ont le même index.

Par exemple la figure 5.2 illustre deux formes (S1) et (S2) d'un chien ainsi que leur décomposition en parties et en lignes de jonction et de disjonction conformément à XLWDOS.

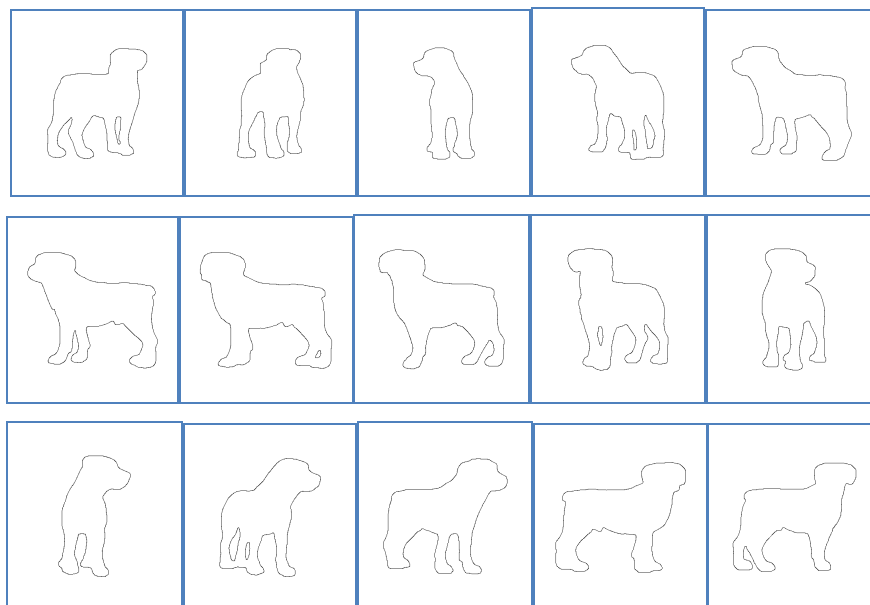


Figure 5.1 Un ensemble de formes de la base d'images de Leibe et Schiele [125]

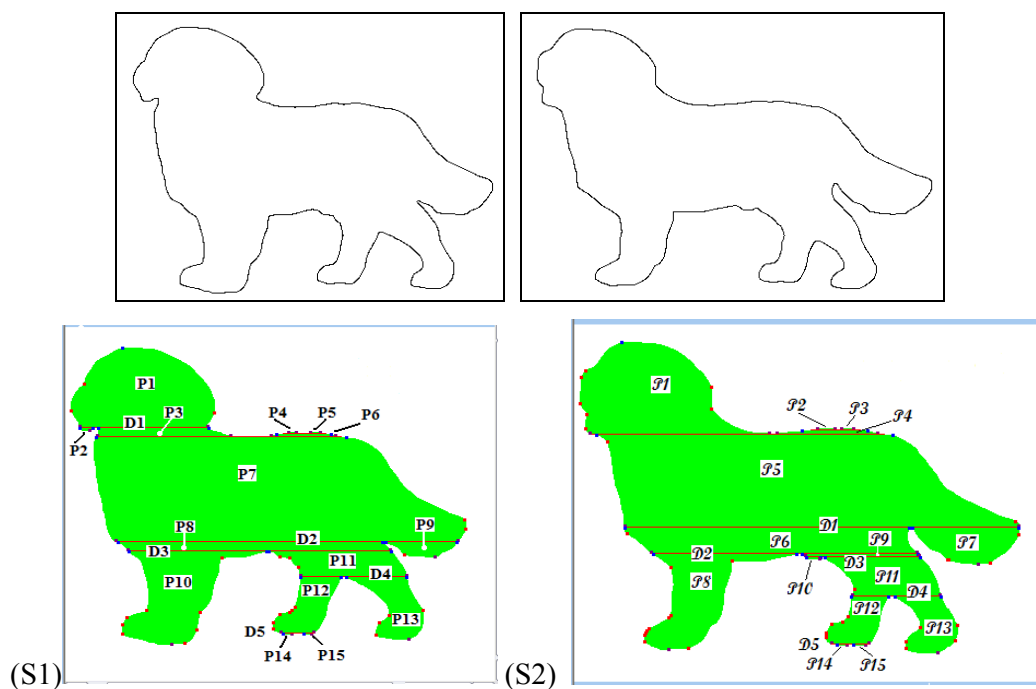


Figure 5.2 Deux images successives d'un chien et leur décomposition en parties.

Les deux descripteurs XLWDOS des formes (S1) et (S2) sont :

(S1) : <CP><CP><CP>P₁ D₁ P₂ P₃</CP> <CP>P₄ P₅ J₁ P₆</CP> J₂ P₇</CP> D₂
 <CP>P₈ D₃ P₁₀ <CP>P₁₁ D₄ <CP>P₁₂ D₅ P₁₄ P₁₅</CP> P₁₃</CP></CP> P₉</CP>

(S2) : <CP><CP>P₁ <CP> P₂ P₃ J₁ P₄ </CP> J₂ P₅ </CP>D₁ <CP>P₆ D₂ P₈<CP>P₉ D₃
 P₁₀ <CP>P₁₁ D₄ <CP>P₁₂ D₅ P₁₄ P₁₅ </CP> P₁₃ </CP></CP> </CP> P₇ </CP>

Les index des formes (S1) et (S2) sont :

(S1) (15, 1, 6, 1011111, 3, 3, 3, 3, 3, 3, 3, 3, jhj&hjhjh&jwj&wjwjw&jwj&jwj&jwj)

(S2) (15, 2, 5, 0110111, 3, 3, 3, 3, 3, 3, 3, jwj&jwj&jwj&jwj&hjwj&jwjw&wjwjw)

Les index sont évidemment différents.

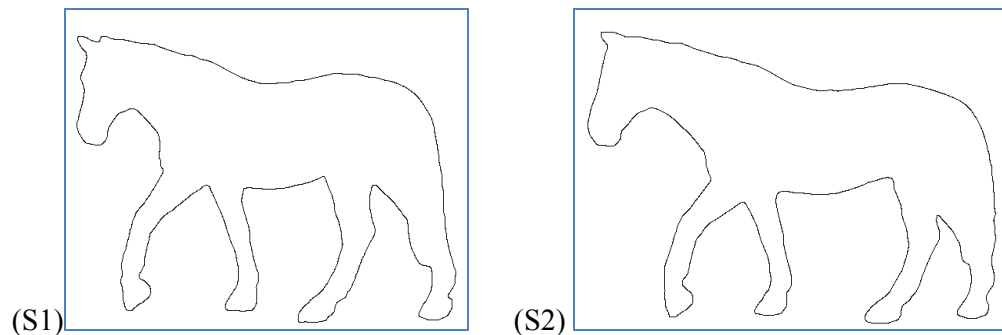


Figure 5.3 Deux images d'un cheval

La figure 5.3 illustre deux images non successives d'un cheval. Le calcul de leurs index a donné les résultats suivants (figure 5.4) :

L'index de la silhouette (S1) est:

(17, 2, 6, 00111111, 3, 3, 3, 3, 3, 3, 3, jwj&jhj&jwj&jhj&jhj&jhjw&jwj&jwjw).

Après le processus de réduction l'index devient:

(12, 2, 4, 001111, 3, 3, 3, 3, jwj&jhj&jwj&jhj&jhj&jhjw).

L'index de la silhouette (S2) est:

(13, 1, 5, 011111, 3, 3, 3, 3, 3, jwj&jhj&jhj&jhj&jhj&jhj).

Les deux formes (S1) et (S2) sont donc différentes, elles ne sont pas mises en correspondance et ne peuvent pas appartenir à la même classe d'objets après l'étape d'indexation.

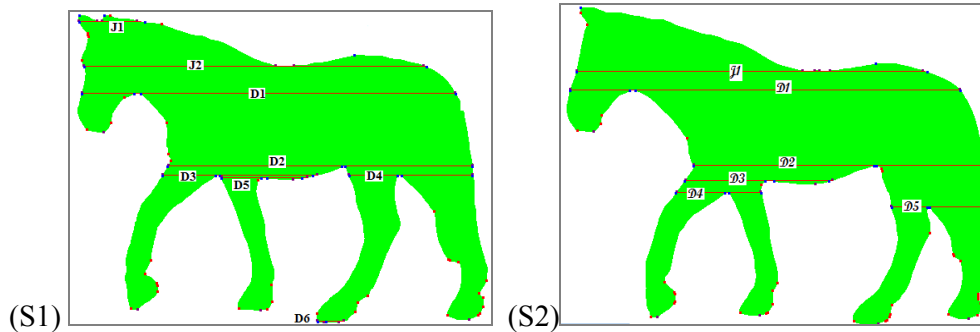


Figure 5.4 Décomposition des images du cheval en parties

5.2.2 Recherche de forme dans la base de modèles par réduction de descripteurs

Nous avons considéré 50% des formes de la base de données de Leibe et Schiele [125] comme des images requêtes et les autres comme modèles. Après plusieurs expérimentations nous avons fixé le seuil $LONG_{SEUIL-MAX}$ à 5 pixels. Les tableaux 5.1 et 5.2 illustrent des exemples des résultants obtenus.

Le tableau 5.1 montre les résultats suivants:

- Lorsque la requête possède le même index que le modèle (par exemple tasse6 avec Tasse1), le processus de l'appariement détaillé sera donc exécuté.
- Pour la requête tasse5 et le modèle tasse2, l'index est retrouvé après avoir appliqué une fois le processus de réduction.
- Pour la requête tasse5 et le modèle tasse3, le processus de réduction est appliqué sans trouver l'index en atteignant le seuil $LONG_{SEUIL-MAX}$.

Les mêmes résultats sont obtenus avec les autres requêtes (tasse6 jusqu'à tasse8), et aussi avec les objets illustrés dans le tableau 5.2.









Modèles \ Requête	 tasse 1	 tasse 2	 tasse 3	 tasse 4
 tasse 5	Même index après 1 itération du processus de réduction	Même index après 1 itération du processus de réduction	Echec de l'appariement	Echec de l'appariement
 tasse 6	Même index sans réduction	Même index sans réduction	Echec d'appariement	Echec de l'appariement
 tasse 7	Echec de l'appariement	Echec de l'appariement	Même index sans réduction	Même index après 1 itération du processus de réduction
 tasse 8	Echec de l'appariement	Echec de l'appariement	Appariement après 1 itération du processus de réduction	Même index sans réduction

Tableau 5.1 Résultats de la recherche d'une forme (Les tasses requêtes sur les lignes et les tasses modèles sur la colonne)

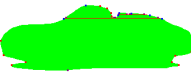


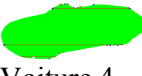


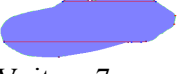

Modèles \ Requête	 Voiture 1	 Voiture 2	 Voiture 3	 Voiture 4
 Voiture 5	Même index sans réduction	Appariement après 2 itérations du processus de réduction	Echec d'appariement	Echec d'appariement
 Voiture 6	Même index après 3 itérations du processus de réduction	Même index après 1 itération du processus de réduction	Echec d'appariement	Echec d'appariement
 Voiture 7	Echec de l'appariement	Echec de l'appariement	Même index après 3 itérations du processus de réduction	Même index après 3 itérations du processus de réduction
 Voiture 8	Echec de l'appariement	Echec de l'appariement	Même index après 2 itérations du processus de réduction	Même index sans réduction

Tableau 5.2. Résultats de la recherche d'une forme (les voitures requêtes sur les lignes et les modèles sur les colonnes)

5.3 Mesures de similarité (calcul de surfaces)

Considérons les tableaux 5.3 et 5.4. Nous remarquons que pour chaque silhouette requête (les images de la ligne du tableau), le modèle correspondant (sur la colonne) est celui dont la différence des surfaces tend vers zéro. En effet dans le cas des images de synthèse la différence est exactement zéro (tableau 5.3) tandis qu'elle est proche de zéro dans le cas d'images réelles (tableau 5.4).









modèle requête				
	0 (1)	5003095,35 (2)	68766171,17 (4)	22115514,91 (3)
	5003095,35 (2)	0 (1)	15710482,46 (4)	12516814,08 (3)
	68766171,17 (4)	15710482,46 (3)	0 (1)	10051295,21 (2)
	22115514,91 (4)	12516968,08 (3)	10051295,21 (2)	0 (1)

Tableau 5.3 Calcul des surfaces dans le cas d'images de synthèse

L'unité de mesure de la surface est le pixel au carré (1pixel = 0.135 mm). Les nombres dans les cases des tableaux sont les différences de surfaces suivies du classement des modèles relativement à leur ressemblance avec la silhouette. Le classement est illustré avec un chiffre entre parenthèses.

5.4 Mesure de similarité (quasi invariants)

Dans cette partie nous reprenons le calcul de surfaces combinées au calcul des quasi-invariants pour la sélection du bon modèle.









modèle requête				
	5,09 (1)	3095,35 (2)	9766131,24 (3)	62766131,70 (4)
	3095.35 (2)	6,32 (1)	3710562,39 (3)	4116827,60 (4)
	9766131,24 (4)	371062,39 (2)	17,98 (1)	514292,37 (3)
	62766131,7 (3)	4116827,6 (2)	6514292,37 (4)	12,03 (1)

Tableau 5.4 Calcul des surfaces dans le cas d'images réelles

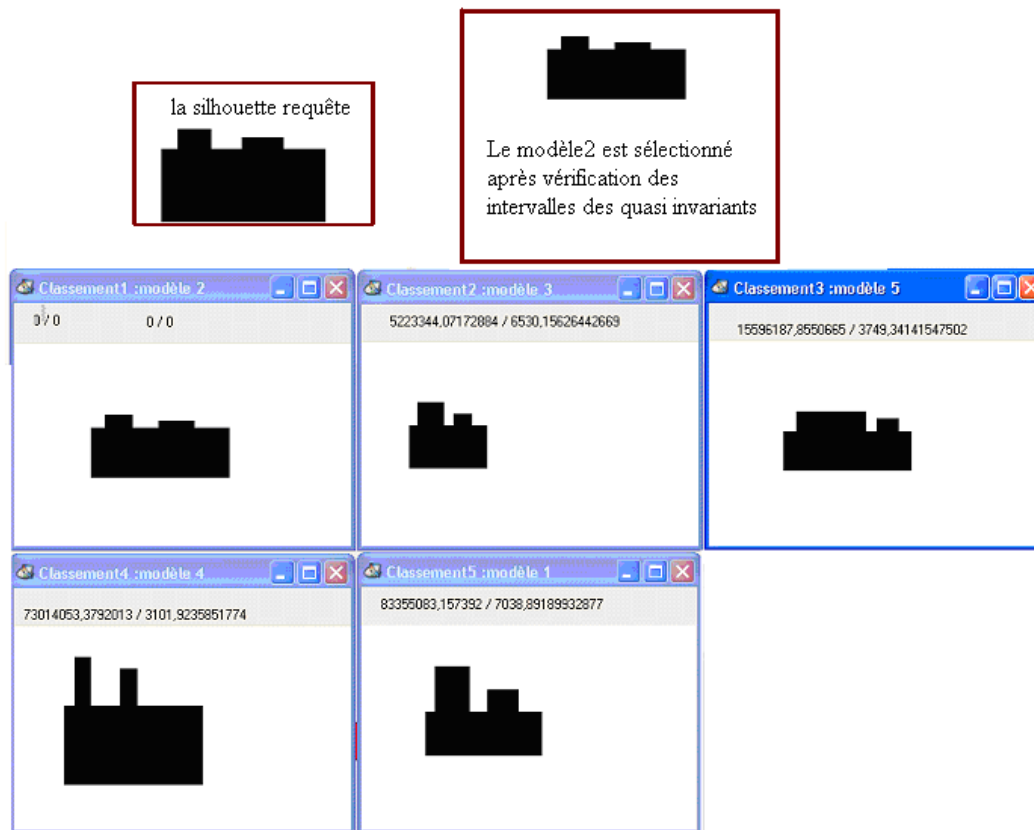


Figure 5.5 Application des seuils sur une image de synthèse et classement des modèles

Les descripteurs détaillés des silhouettes de la figure 5.5 sont :

La requête:

```
<CP>
<P1><L> R 90 28 </L><R> R 180 43 R 90 28 </R></P1>
<P2><L> R 90 16 </L><R> R 180 56 R 90 16 </R></P2>
<J1> W P3 25 S P1 P3 43 W P3 49 S P2 P3 56 W P3 61 </J1>
<P3><L> R 90 99 R 180 234 </L><R> R 90 99 </R></P3>
</CP>
```

Le modèle 1 :

```
<CP>
<P1><L> R 90 84 </L><R> R 180 53 R 90 84 </R></P1>
<P2><L> R 90 41 </L><R> R 180 48 R 90 41 </R></P2>
<J1> W P3 14 S P1 P3 53 W P3 26 S P2 P3 48 W P3 36 </J1>
<P3><L> R 90 82 R 180 177 </L><R> R 90 82 </R></P3>
</CP>
```

Le modèle 2 :

```
<CP>
<P1><L> R 90 25 </L><R> R 180 38 R 90 25 </R></P1>
<P2><L> R 90 14 </L><R> R 180 50 R 90 14 </R></P2>
<J1> W P3 22 S P1 P3 38 W P3 43 S P2 P3 50 W P3 54 </J1>
<P3><L> R 90 88 R 180 207 </L><R> R 90 88 </R></P3>
</CP>
```

Le modèle 3 :

```
<CP>
<P1><L> R 90 44 </L><R> R 180 39 R 90 44 </R></P1>
```

```
<P2><L> R 90 23 </L><R> R 180 27 R 90 23 </R></P2>  
<J1> W P3 12 S P1 P3 39 W P3 14 S P2 P3 27 W P3 23 </J1>  
<P3><L> R 90 81 R 180 115 </L><R> R 90 81 </R></P3>  
</CP>
```

Le modèle 4 :

```
<CP>  
<P1><L> R 90 93 </L><R> R 180 25 R 90 93 </R></P1>  
<P2><L> R 90 70 </L><R> R 180 25 R 90 70 </R></P2>  
<J1> W P3 16 S P1 P3 25 W P3 45 S P2 P3 25 W P3 100 </J1>  
<P3><L> R 90 148 R 180 211 </L><R> R 90 148 </R></P3>  
</CP>
```

Le modèle 5:

```
<CP>  
<P1><L> R 90 37 </L><R> R 180 107 R 90 37 </R></P1>  
<P2><L> R 90 24 </L><R> R 180 33 R 90 24 </R></P2>  
<J1> W P3 18 S P1 P3 107 W P3 16 S P2 P3 33 W P3 21 </J1>  
<P3><L> R 90 75 R 180 195 </L><R> R 90 75 </R></P3>  
</CP>
```

La figure 5.5 montre aussi le classement des modèles par ordre de mérite (par ordre décroissant de la mesure de similarité calculée sur les différences des surfaces).

- Comparaison de la requête avec le modèle1 :
La différence des surfaces calculée est : 83355083,16
- Comparaison de la requête avec le modèle2 :
La différence des surfaces calculée est : **0**
- Comparaison de la requête avec le modèle3 :
La différence des surfaces calculée est : 5223344,07

- Comparaison de la requête avec le modèle4 :
La différence des surfaces calculée est : 73014053,38
- Comparaison de la requête avec le modèle5 :
La différence des surfaces calculée est : 15596187,85.

Pour tester les quasi-invariants, nous avons considéré les descriptions détaillés (données ci-dessus) de tous les modèles pris deux à deux avec la requête. Seul le modèle2 vérifie les seuils des quasi invariants ($\ln(p)=0.23$, $\theta=18.61^\circ$) d'où sa sélection comme étant le modèle recherché.

Le même traitement est effectué sur l'image réelle d'une voiture (figure 5.6)

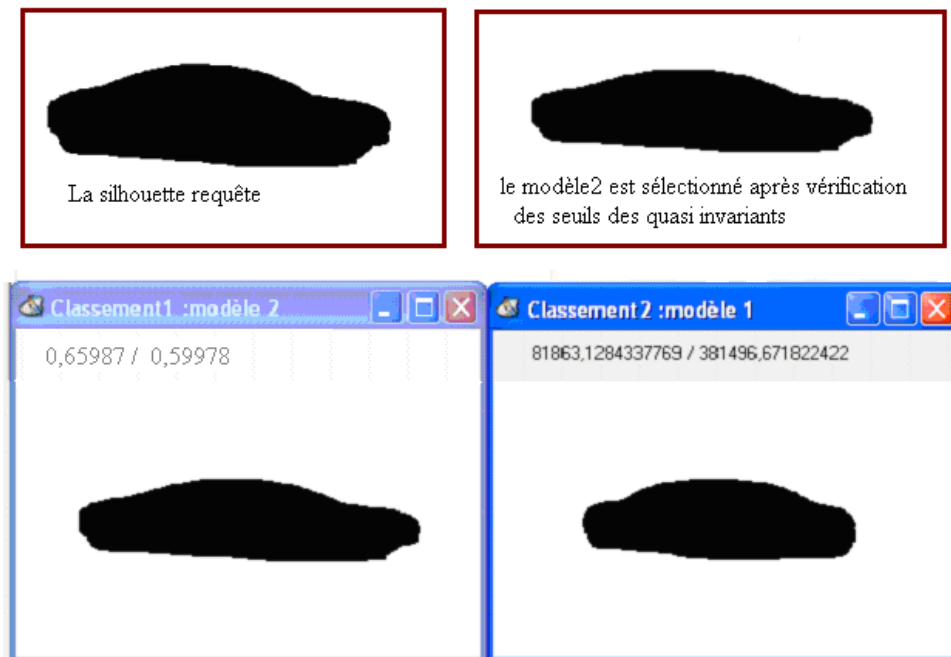


Figure 5.6 Application des seuils sur une image réelle et classement des modèles

Les descripteurs détaillés des silhouettes de la figure 5.6 sont :

La requête:

<CP>
 <CP>
 <P2><L> CC 5 8 5 </L><R> CV 2 177 6 </R></P2>
 <P1><L> CC 3 8 5 </L><R> R 152 5 </R></P1>
 <J1> W P3 5 S P2 P3 82 W P3 17 S P1 P3 45 W P3 3 </J1>
 <P3><L> CV 33 58 86 CV 10 149 25 </L><R> CV 15 156 55 CV 18 59 56
 </CP>
 <D1> H P3 10 S P3 P4 105 H P3 33 S P3 P5 98 H P3 5 </D1>
 <P4><L> CV 7 172 5 </L><R> CC 3 3 5 </R></P4>
 <P5><L> CC 3 172 8 </L><R> CC 9 16 8 </R></P5>
</CP>

Le modèle 1 :

<CP>
 <CP>
 <P2><L> CC 5 8 4 </L><R> CV 2 177 4 </R></P2>
 <P1><L> CC 3 6 5 </L><R> R 150 5 </R></P1>
 <J1> W P3 5 S P2 P3 80 W P3 17 S P1 P3 44 W P3 4 </J1>
 <P3><L> CV 4 21 20 CV 5 91 66 </L><R> CV 15 156 55 CV 8 75 39 </R></P3>
 </CP>
 <D1> H P3 4 S P3 P4 151 H P3 8 S P3 P5 90 H P3 4 </D1>
 <P4><L> CV 12 172 3 </L><R> CC 10 3 12 </R></P4>
 <P5><L> CC 3 172 8 </L><R> CC 23 16 35 </R></P5>
</CP>

Le modèle2 :

<CP>
 <CP>
 <P2><L> CC 5 8 4 </L><R> CV 2 177 4 </R></P2>
 <P1><L> CC 3 8 5 </L><R> R 150 5 </R></P1>

```

<J1> W P3 5 S P2 P3 80 W P3 17 S P1 P3 44 W P3 4 </J1>
<P3><L> CV 33 58 86 CV 10 149 25 </L><R> CV 15 156 55 CV 18 59 56
</R></P3>
</CP>
<D1> H P3 10 S P3 P4 106 H P3 36 S P3 P5 98 H P3 5 </D1>
<P4><L> CV 7 172 5 </L><R> CC 3 3 5 </R></P4>
<P5><L> CC 3 175 8 </L><R> CC 9 16 8 </R></P5>
</CP>

```

- Comparaison de la requête avec le modèle1 :
La différence des surfaces calculée est : 81863,13
- Comparaison de la requête avec le modèle2 :
La différence des surfaces calculée est : 0,65987

Le modèle2 vérifie les seuils des quasi invariants ($\ln(\rho)=0.23$, $\theta=18.61^\circ$), il minimise aussi la différence de surface avec la requête, d'où sa sélection comme étant le modèle recherché.

D'autres exemples d'expérimentation sur des images réelles sont présentés dans les tableaux ci-dessous (les tableaux 5.5, 5.6 et 5.7). Nous utilisons quelques images construites par B. Leibe et B. Schiele [125] où chaque objet est représenté par 41 vues différentes. Nous montrons pour chaque cas le modèle sélectionné. Dans le cas où plusieurs modèles vérifient les seuils des quasi invariants, le modèle qui nous intéresse sera celui qui minimise la différence des surfaces avec la requête, il sera symbolisé dans le tableau par la lettre R (qui signifie reconnu). Les modèles vérifiant les seuils de quasi-invariants seront soulignés. Les silhouettes sur les colonnes représentent les modèles sélectionnés de toutes les étapes antérieures du traitement, (qui résultent de l'étape de l'indexation), et les silhouettes sur les lignes des tableaux correspondent aux requêtes. Là aussi les mesures sont données en pixel au carré.








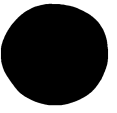
Modèles Requêtes				
	<u>782,66</u>	<u>19,92</u> (R)	<u>3271,56</u>	1733,17
	<u>7679,92</u>	<u>1237,94</u>	97539,96	<u>38,59</u> (R)
	<u>6521,56</u>	<u>53,95</u> (R)	76758,43	<u>12384,49</u>
	2369,167111	36977,99	<u>18,62</u> (R)	234,67

Tableau 5.5 Reconnaissance des images d'une pomme









Modèles Requêtes				
	4358987,65	435529,30	<u>24,99</u> (R)	<u>271,62</u>
	456565,29	785433,63	<u>46,79</u> (R)	<u>8559,75</u>
	3634564,93	942625,49	4356,17	<u>38,84</u> (R)
	127710,62	128559,75	827208,85	<u>12,06</u> (R)

Tableau 5.6 Reconnaissance des images d'une poire









Modèles \ Requête				
	982139754,33	439814563,64	1376789227,17	521,95
	6123,14	<u>13,67</u> (R)	135817,96	51239765,38
	1379227,17	13585417,96	4598762,88	<u>21,86</u> (R)
	596542121,95	526789465,38	67818939,80	289754,38

Tableau 5.7 Reconnaissance des images d'une tasse (avec des cas d'échec de reconnaissance)

Nous remarquons dans le tableau 5.7 que la première et la dernière requête ne sont pas reconnues. Elles ne vérifient pas les seuils des quasi-invariants avec les modèles de la base.

5.5 Autres expérimentations

Nous récapitulons dans ce paragraphe l'application de l'algorithme de reconnaissance sur deux bases de données qui sont la base de [125] et la base de Mokhtarian et Al [148].

5.5.1 Application sur la base d'images de Leibe et AL

La base de Leibe [125] est constituée de 80 objets (10 pommes, 10 vaches, etc..) dont les types sont montrés dans la figure 5.7. Pour chacun de ces 80 objets, 41 vues différentes sont obtenues. Ceci nous donne au total un nombre de 3280 images différentes dans la base. Par exemple la figure 5.8 montre toutes les vues de l'un des objets (vache).

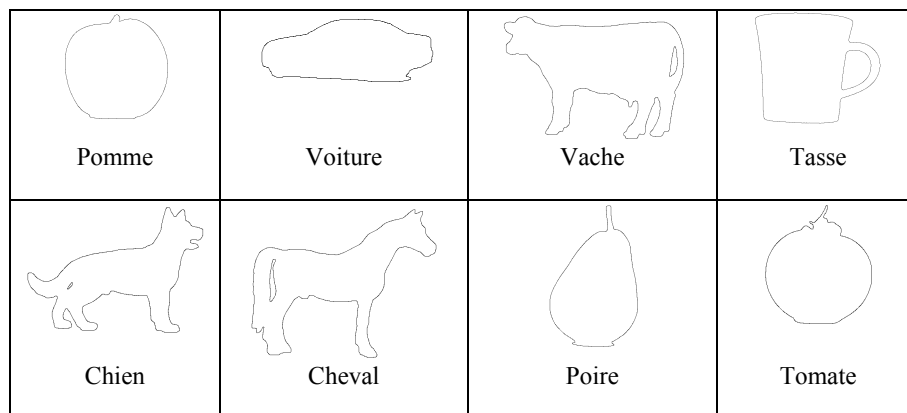


Figure 5.7 Les types d'objets dans la base [125]

La figure 5.9 montre le résultat de l'application de l'algorithme de reconnaissance. Tous les objets ayant le même index que la requête sont sélectionnés (par l'algorithme) de la base d'images et sont représentés dans la même figure. Ces index sont obtenus à la suite de l'application de l'algorithme de lissage et celui d'indexation.

Les modèles sélectionnés à la suite de l'indexation sont donnés par ordre de mérite selon le calcul de la différence des surfaces avec la requête (la différence des surfaces notée DF est indiquée pour chaque modèle). Le modèle ayant l'étiquette **R** (qui signifie reconnu) est celui qui vérifie les seuils des quasi-invariants et qui est retenu par l'algorithme comme étant l'objet modèle qui ressemble le plus à la requête.

Ces résultats nous montrent l'efficacité de l'algorithme d'indexation qui a considérablement réduit le nombre d'images qui restent pour l'étape de la reconnaissance c'est-à-dire l'étape appliquant les mesures de similarité pour trouver le modèle recherché.

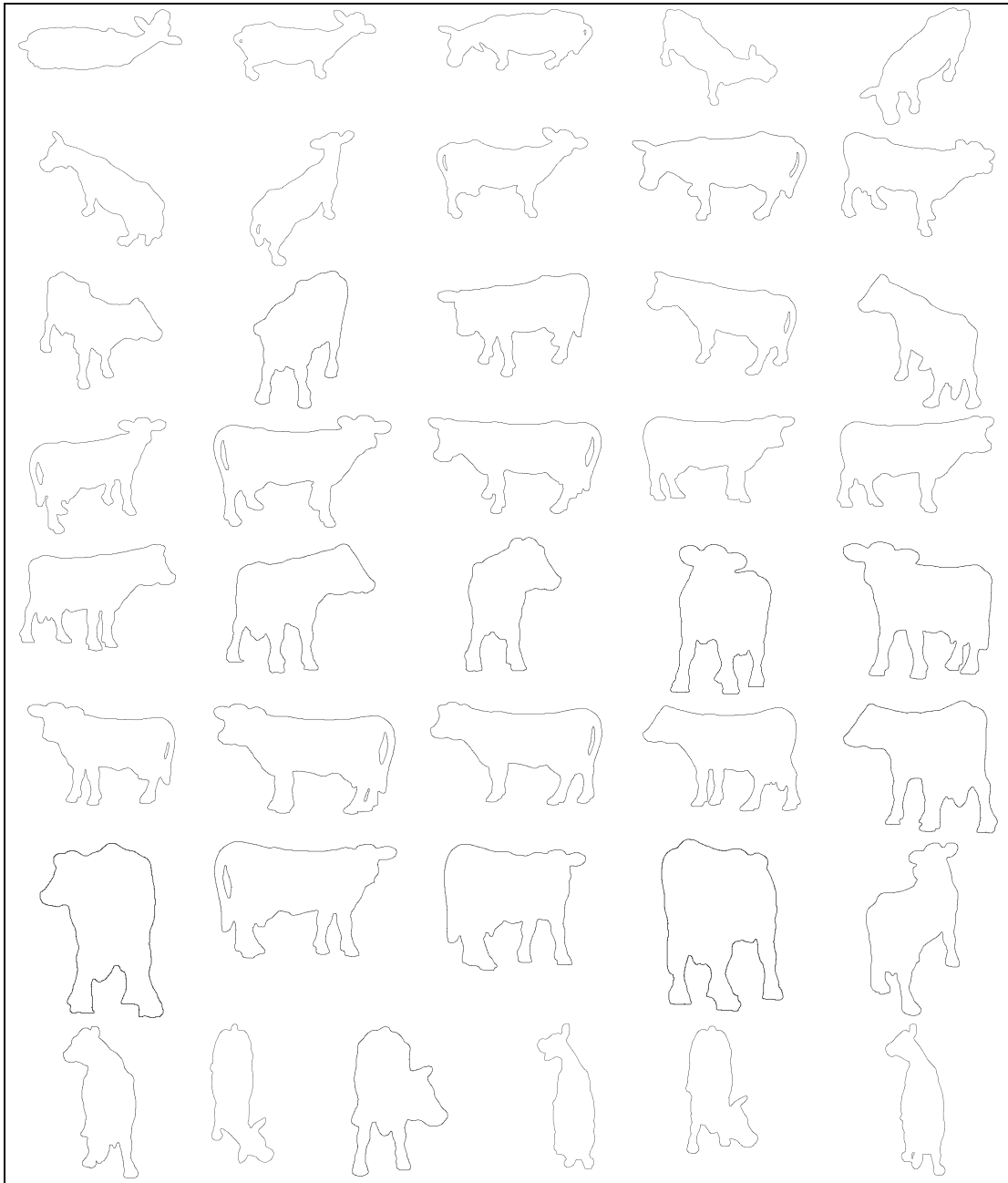


Figure 5.8 Les 41 vues d'un des objets de la base de [125]

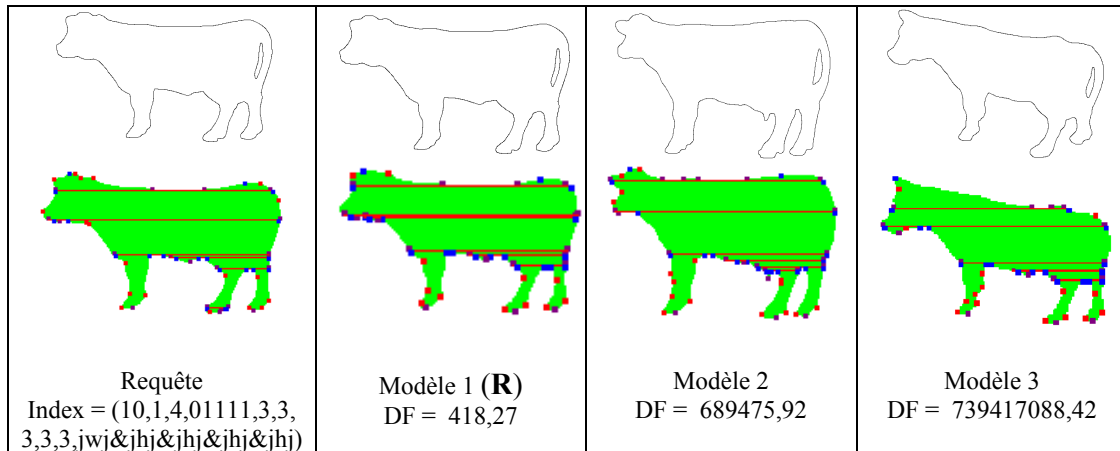


Figure 5.9 Reconnaissance d'un objet (vache)

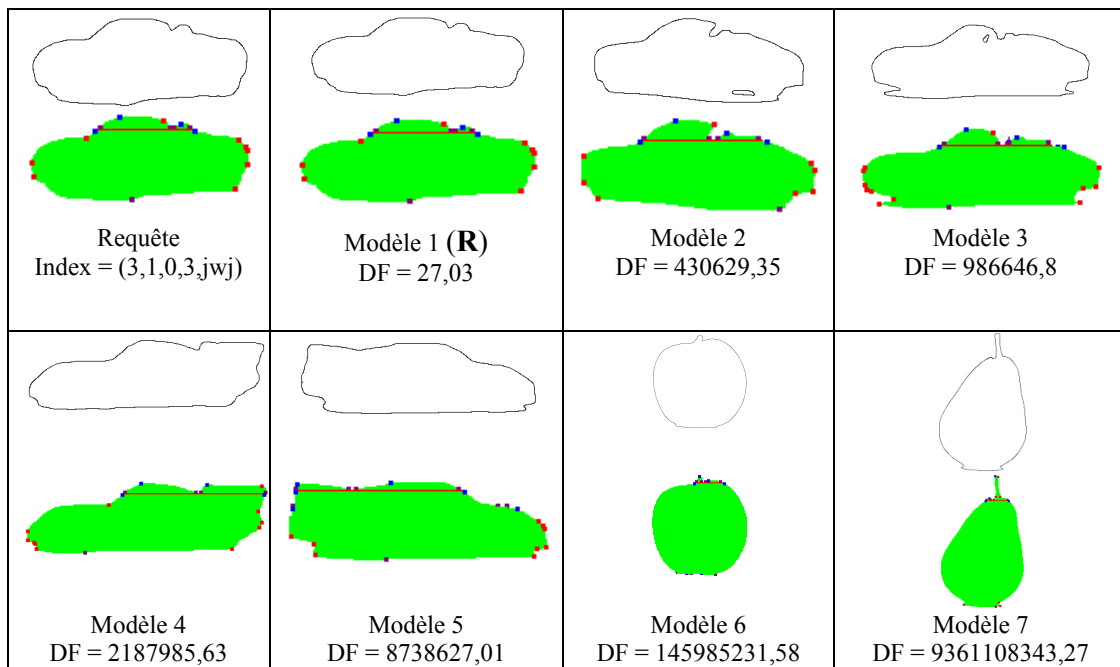


Figure 5.10 Reconnaissance d'un objet (voiture)

La figure 5.10 illustre un autre exemple d'expérimentation, les images ayant le même index que la voiture requête sont sélectionnées. Nous remarquons que ce nombre est légèrement supérieur au cas précédent ceci est expliqué par le fait que l'objet (vache) est plus compliqué dans sa structure que l'objet (voiture). En effet quand l'objet a une structure complexe peu d'images peuvent être sélectionnés après l'indexation vu la nature des index utilisés.

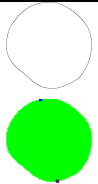





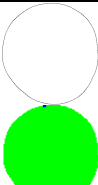
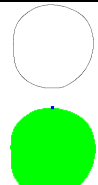



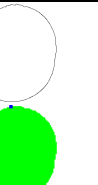
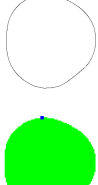





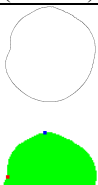
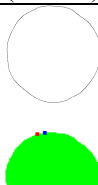
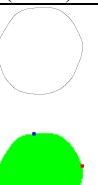



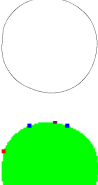



 Requête(Pomme) Index = (1,0,0)	 Modèle 1 (R) DF = 6,24 (Poire)	 Modèle 2 DF = 69,32 (Pomme)	 Modèle 3 DF =94,98 (Pomme)	 Modèle 4 DF =175,28 (Pomme)	 Modèle 5 DF =208,54 (Tomate)
 Modèle 6 DF =307,81 (Tomate)	 Modèle 7 DF =397,68 (Poire)	 Modèle 8 DF =1298,9 (Pomme)	 Modèle 9 DF =1355,99 (Pomme)	 Modèle 10 DF =2199,31 (Pomme)	 Modèle 11 DF =3181,82 (Poire)
 Modèle 12 DF =3725,66 (Tomate)	 Modèle 13 DF =10709,86 (Tomate)	 Modèle 14 DF=10736,66 (Poire)	 Modèle 15 DF=11325,91 (Poire)	 Modèle 16 DF=11362,11 (Poire)	 Modèle 17 DF=12276,76 (Tomate)
 Modèle 18 DF=12335,26 (Poire)	 Modèle 19 DF=12922,55 (Tomate)	 Modèle 20 DF=13010,94 (Tomate)	 Modèle 21 DF=24269,3 (Tomate)	 Modèle 22 DF=24327,96 (Poire)	 Modèle 23 DF=30126,2 (Tomate)
 Modèle 24 DF=46023,4 (Pomme)	 Modèle 25 DF=46602,07 (Pomme)	 Modèle 26 DF=47462,9 (Pomme)	 Modèle 27 DF=258251,36 (Poire)		

Figure 5.11 Cas d'échec de l'algorithme de reconnaissance pour une vue non caractéristique.

La figure 5.11 montre un cas d'échec de l'algorithme de reconnaissance. Nous constatons que pour l'objet requête (pomme) beaucoup d'objets différents sont sélectionnés à la suite de l'indexation. Un modèle de type (poire) est trouvé comme celui qui correspond à la requête après l'application des mesures de similarité. L'échec de l'algorithme dans ce cas est dû à la vue non caractéristique de la requête considérée. En effet même le système visuel humain avec toutes ses performances est incapable de dire que la requête est une pomme ou autre chose (poire, tomate, ballon, etc...), c'est pour cette raison d'ailleurs que nous avons préféré indiquer pour chaque image de la figure 5.11 le type de l'objet représenté.

5.5.2 Application sur la base de Mokhtarian

La base de donnée de Mokhtarian [148] est constituée de 1100 images contours de différents types de poissons. Des exemples d'images de cette base sont donnés par la figure 5.12.

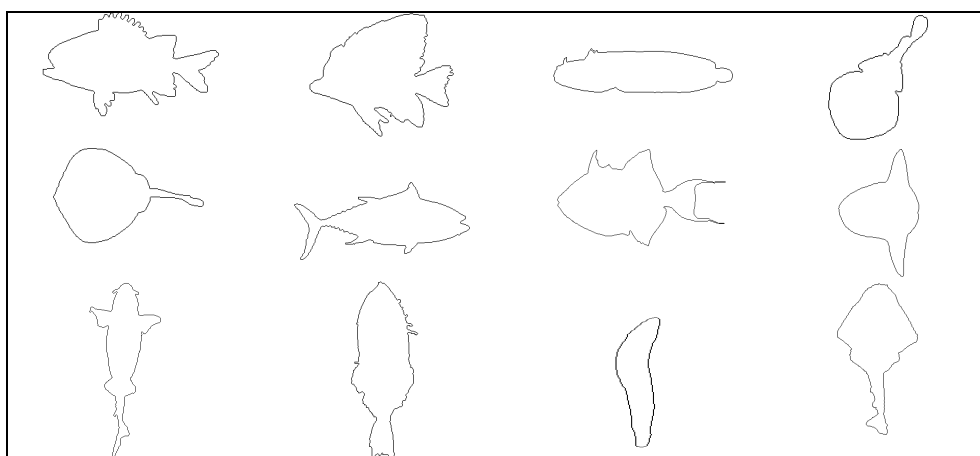


Figure 5.12 Exemples de types d'images de la base [148]

Comme le cas de la base de Leibe et AL [125], le modèle reconnu est indiqué par la lettre (R). Les modèles ayant le même index sont classés par ordre croissant de la différence de surface avec la requête.

Peu de modèles sont trouvés dans le cas de la figure 5.13 où la structure de l'index est assez complexe, contrairement à la figure 5.14 où les objets sont constitués d'une seule partie selon XLWDOS, on a pu trouver un nombre plus élevé de modèles à l'issue de l'indexation mais un seul est reconnu après avoir appliqué les mesures de similarité.

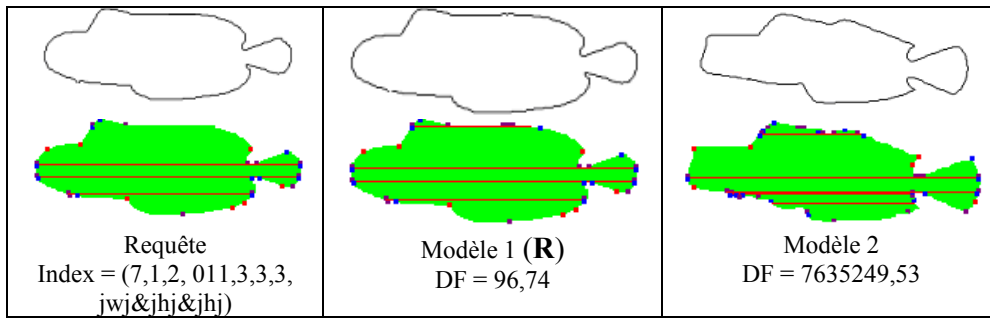


Figure 5.13 Reconnaissance dans le cas d'un index complexe pour la base [148]

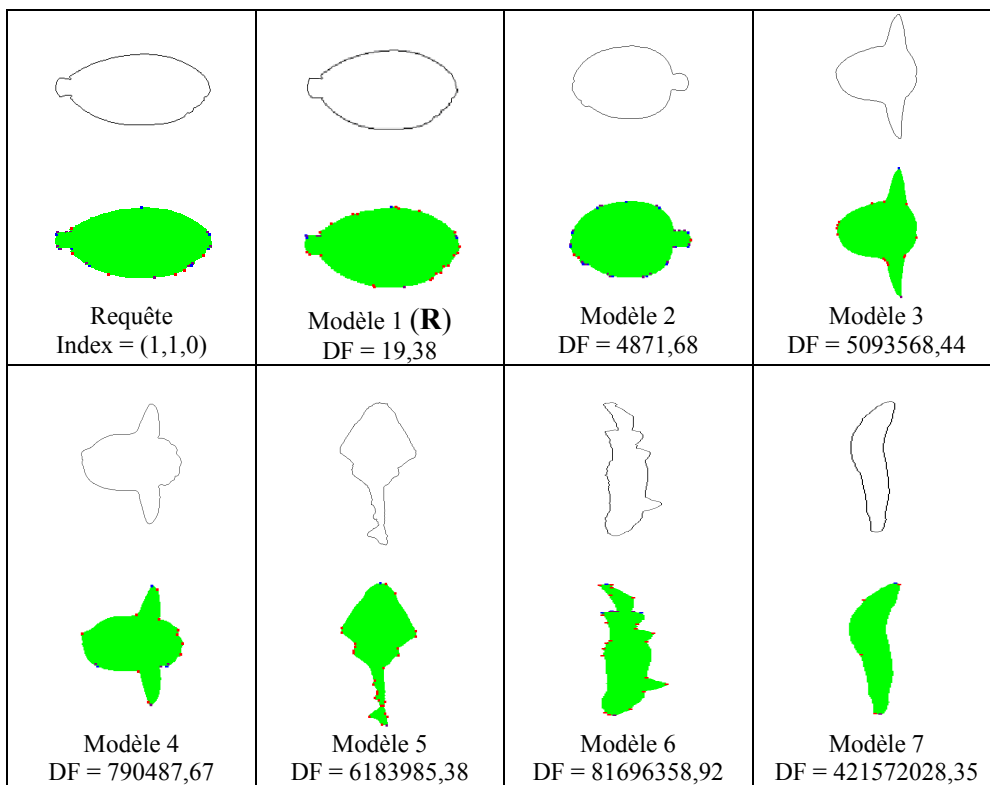


Figure 5.14 Reconnaissance dans le cas d'un index simple pour la base [148]

Considérons maintenant la figure 5.15. Nous constatons qu'aucun modèle n'est reconnu après l'application des mesures de similarité, ceci est dû à l'absence d'un modèle dont la prise de vue est très proche de celle de la requête.

















  Requête Index = (1,1,0)	  Modèle 1 DF = 293784,72	  Modèle 2 DF = 956301,68	  Modèle 3 DF = 2983451,83
  Modèle 4 DF = 5921693,44	  Modèle 5 DF = 16183985,38	  Modèle 6 DF = 81696358,92	  Modèle 7 DF = 624583068,35

Figure 5.15 Absence du modèle similaire à la requête dans la base [148]

5.6 Conclusion

Nous avons validé dans ce chapitre notre algorithme de reconnaissance.

Nous avons commencé par valider l'algorithme d'indexation en montrant qu'il est possible d'obtenir le même index (si les index d'origine sont différents) lorsque les points de la prise de vue des images sont proches et ce en appliquant l'algorithme de réduction. Nous avons montré aussi que cette technique nous permet d'indexer une base d'images et de rechercher ensuite une silhouette modèle ayant le même index que la requête. Seuls les modèles restants de cette étape vont être concernés par les traitements de comparaison.

Le calcul de surfaces a montré l'efficacité de l'algorithme pour classer les modèles, car lorsque le modèle ressemble à la requête, il se voit attribué une différence de surface qui tend vers zéro lorsqu'il est pris avec la requête.

Cependant l'utilisation de différence de surfaces est une condition nécessaire mais pas suffisante pour sélectionner les « bons » modèles. Pour cette raison un système de reconnaissance complet doit prendre en considération la géométrie de chaque partie en plus de sa surface et ce en calculant des mesures de similarité pour les descripteurs détaillés. Pour ce faire et cerner le modèle recherché nous avons appliqué une mesure de similarité basée sur les quasi invariants géométriques afin de garantir une grande ressemblance quant à la description détaillée entre la requête et le modèle recherché.

Conclusion Générale et Perspectives

Conclusion

Comme nous l'avons présenté dans cette thèse, la recherche en vision par ordinateur se répartit en deux grandes classes. Ce sont l'approche classique dite *vision orientée modèle* (appelée aussi *approche basée objet*) et la *vision par apparence* qui s'appuie sur toutes les manifestations visuelles possibles des entités observées. Cette dernière permet de réduire la tâche de la reconnaissance à un problème de mise en correspondance entre deux images (une image requête et une image modèle), et ne se base sur aucune modélisation 3D. Le principe de la perception par apparence est de projeter l'espace d'apparences dans un espace de dimensionnalité réduite choisi de façon à faciliter les tâches à accomplir (indexation, reconnaissance, estimation de la position, etc.)

Parmi les méthodes basées sur l'apparence, on distingue celles qui exploitent les silhouettes d'objets, entre autre la méthode LWDOS/XLWDOS développée au laboratoire LRIA de l'U.S.T.H.B [120, 121] et qui a constitué le point de départ de notre travail.

Les descripteurs obtenus de cette approche peuvent changer dans leurs structures à cause de l'existence du bruit ou à cause d'un léger changement dans la prise de vue des images, ce qui rendrait une comparaison « brute » des descripteurs une opération difficile. Pour cette raison l'utilisation directe de tels descripteurs devient impossible dans le cadre de la reconnaissance d'objets réels. Pour pallier à ce problème, nous avons proposé des algorithmes qui tiennent compte de telles déformations pour la recherche d'une forme requête dans une base de descripteurs modèles.

Nous avons commencé par proposer une méthode permettant l'appariement des descripteurs globaux selon LWDOS et ce en développant un algorithme de réduction de descripteurs qui garantit un niveau de souplesse et de tolérance dans le processus de mise en

correspondance. Ensuite nous nous sommes intéressées à l'approximation des descripteurs détaillés en appliquant un ensemble de règles que nous avons définies.

Les résultats sont satisfaisants mais ils sont limités à une comparaison grossière des formes car LWDOS décrit les formes avec leurs dimensions relatives. L'intérêt majeur de cette première étude était de tester la faisabilité de la comparaison de descripteurs en présence du bruit avant de passer au processus de reconnaissance. On peut appliquer cette méthode dans le cadre de la vision et de la reconnaissance grossière des formes mais non pas dans des applications multimédia nécessitant de connaître les dimensions exactes.

Nous sommes ensuite passés à l'utilisation de la description XLWDOS qui elle, s'intéresse aux dimensions exactes. Nous avons commencé par l'indexation de descripteurs afin de réduire le nombre de modèles qui seront concernés par l'application des mesures de similarité.

Nous avons proposé deux méthodes pour la recherche de l'index d'une forme requête dans la table des index des descripteurs modèles. Ces méthodes tiennent compte des déformations pouvant altérer la géométrie de la frontière de la silhouette requête. La première méthode consiste à appliquer un ensemble de réductions au descripteur requête afin de trouver un ensemble de modèles ayant le même index. La seconde opère un lissage sur la silhouette requête afin de réduire l'effet des déformations.

A l'issue de l'indexation, le nombre de modèles qui restent pour la reconnaissance est considérablement réduit.

La mesure de similarité des descripteurs globaux est basée sur la différence des surfaces des parties prises entre la requête et les modèles. Le traitement a permis de classer les modèles selon l'ordre décroissant de ressemblance.

Le recours aux différences de surfaces a permis aussi de filtrer d'avantage les silhouettes modèles et ne garder que celles qui minimisent les différences des aires des surfaces des parties prises deux à deux entre la silhouette requête et la silhouette modèle.

Etant donné que l'utilisation des surfaces n'est pas une condition suffisante pour sélectionner le bon modèle, nous avons alors opté pour l'étude de la géométrie de chaque

élément XLWDOS c'est-à-dire à la ressemblance des descripteurs détaillés. Pour cela nous nous sommes basés sur l'utilisation des quasi-invariants géométriques (l'angle entre deux segments et leur rapport de distance) ces mesures ne varient pas d'une manière considérable lorsque les images sont proches. Pour trouver les meilleurs seuils une étude expérimentale a été effectuée.

Perspectives

Comme perspectives à notre travail nous proposons de résoudre les problèmes suivants :

- Adapter nos algorithmes au cas de silhouettes d'objets présentant des formes internes. En effet, une autre version de la description textuelle a été développée par notre équipe et qui tient compte des détails internes.
- Etudier la possibilité d'appliquer notre approche à des objets en occlusion. La reconnaissance des objets en occlusion est un problème loin d'être résolu. Comme notre méthode réduit les descripteurs initiaux en sous descripteurs, nous essayons dans ce cas de l'appliquer à ce type d'objets qui verront par conséquent une partie de leurs descriptions tronquée à cause de l'occlusion.
- Utiliser cette technique dans des applications multimédia et la recherche d'images par le contenu, notamment que la représentation textuelle devrait accélérer la recherche.

Annexe A

Lissage gaussien

A.1 Convolution avec le filtre gaussien

La valeur en niveau de gris (valeur comprise entre 0 et 255) d'un pixel est recalculée en fonction de celle de ses huit voisins qui l'entourent.

Le filtre gaussien est décrit par une matrice carrée appelée matrice de convolution dont les éléments sont calculés en fonction de la position des pixels et la variance σ qui est un paramètre à choisir. Cette matrice est calculée pour chaque pixel de position (x,y) de la silhouette selon la formule suivante :

$$\text{MatConvolution}(x,y,\sigma) = (1 / \sigma \sqrt{2\pi}) e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

Une fois la matrice de convolution définie, on passe à la convolution qui n'est qu'un produit matriciel entre la matrice de l'image et la matrice de convolution. Enfin la nouvelle valeur d'intensité d'un pixel en position (x,y) est définie par :

$$\text{MatFiltré}(x,y) = \sum_{i=-1}^{i=1} \sum_{j=-1}^{j=1} \text{MatImage}(x+i,y+j) * \text{MatConvolution}(i,j).$$

Pour filtrer toute l'image on applique l'algorithme du filtre gaussien:

Algorithme Filtre_Gaussien**DEBUT**

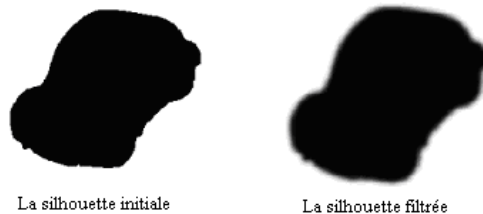
/*entrée : MatImage ; la matrice image initiale

/*sortie : MatFiltré ; la matrice après filtrage

Donner σ ;**Pour** chaque pixel de MatImage **Faire**

$$\text{MatConvolution}(x,y,\sigma) = (1/\sigma\sqrt{2\pi})e^{-\frac{(x^2+y^2)}{2\sigma^2}} ;$$

$$\text{MatFiltré}(x,y) = \sum_{i=-1}^{i=1} \sum_{j=-1}^{j=1} \text{MatImage}(x+i,y+j) * \text{MatConvolution}(i,j) ;$$

Fait.**FIN.**Figure A.1 Application du filtre gaussien ($\sigma=2$)

La figure A.1 montre une silhouette et le résultat obtenu après l'application du filtre gaussien. On peut constater que l'application de ce filtre rend l'image floue, d'où la nécessité de passer à une extraction de contours après le lissage.

A.2 Extraction de contours

La figure A.2 illustre la détection de contours sur la silhouette de la figure précédente préalablement filtrée en utilisant un seuillage.

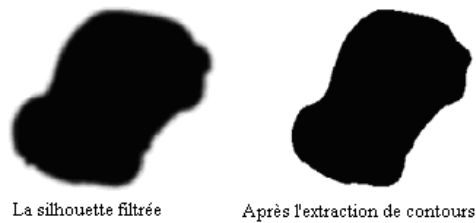


Figure A.2 extraction de contours (seuil =128)

L'effet flou dû au filtrage est expliqué par l'occurrence des pixels gris dans l'image à la suite de la convolution, pour lesquels on appliquera un algorithme d'extraction de contour afin de les éliminer. Le but de l'algorithme de seuillage est d'éliminer les pixels qui possèdent un niveau de gris inférieur au seuil que nous avons fixé à 128. Ce choix du seuil est défini après plusieurs expériences et essai de plusieurs valeurs.

Si la valeur du pixel traité est supérieure à 128 alors il sera recoloré en blanc et le pixel n'appartiendra plus à la silhouette. Sinon, c'est à dire que la valeur du pixel est inférieure ou égale au seuil, nous le recolorons en noir et il est considéré appartenant à la silhouette.

L'application de ce seuillage permet d'obtenir une image moins bruitée avec des contours plus lisses.

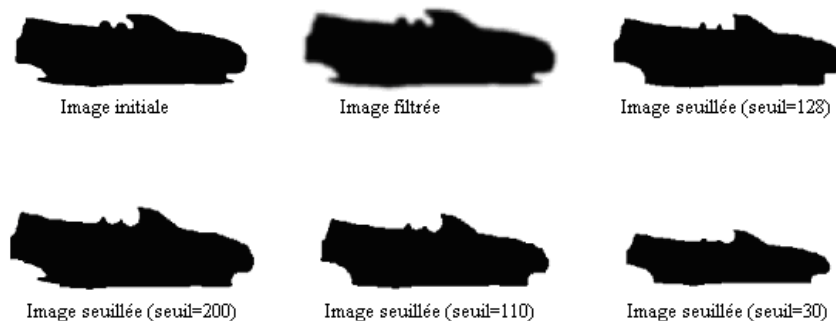


Figure A.3 extraction de contours avec différentes valeurs du seuil

La figure A.3 illustre l'une des expérimentations effectuées dans le but de fixer le bon seuil. Nous pouvons remarquer qu'une valeur supérieure à 128 laisse beaucoup de détails sur le contour qui n'est pas très différent du contour initial. Les valeurs inférieures à 128 ont tendance par contre à trop rétrécir le contour.

Le but de l'application de l'algorithme de lissage et de l'extraction de contours, est l'élimination de toutes les parties dues au bruit.

Après l'application de cet algorithme pour une valeur σ , le descripteur ne contiendra plus de parties bruitées[119].

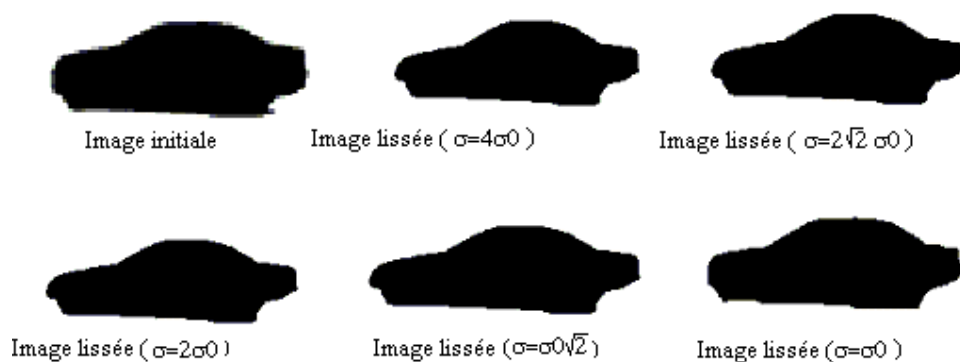


Figure A.4 lissage avec différentes valeurs de σ ($\sigma_0 = 1.3$)

La figure A.4 montre l'application de l'algorithme de lissage sur la silhouette d'une voiture avec différentes valeurs de σ . Pour l'extraction de contours après le lissage gaussien nous avons maintenu la valeur du seuil à 128.

Annexe B

Calcul des surfaces des parties XLWDOS

B.1 Calcul de la surface d'une partie

Pour calculer la surface d'une partie, nous appliquons la formule des trapèzes.

Le calcul de la surface de la partie XLWDOS (voir figure B.1.a) se divise en deux : la surface de la partie gauche délimitée par la frontière gauche et la surface de la partie droite délimitée par la frontière droite (voir figure B.1.b). Ces parties sont délimitées par une droite commençant du point le plus haut à gauche et ayant la direction des colonnes de l'image

Pour chacune des frontières de la partie, la première étape consiste à déterminer les points de début et de fin des ses contours simples. Soient $P_0^g, P_1^g, P_2^g, \dots, P_n^g$ ces points d'extrémités.

Commençant à partir du point P_0^g , il s'agit de trouver la suite de points $P_0^g, P_1^g, P_2^g, \dots, P_n^g$ dont les abscisses suivant le repère de travail OX sont ordonnées et de calculer l'aire comprise entre la frontière et la droite séparant les deux frontières.

Si n n'est pas le dernier point, donc P_{n+1}^g est un point dont l'abscisse est en désordre avec la liste des points précédents. Nous déterminons la suite de points ayant l'ordre de P_{n+1}^g et calculons l'aire définie par ces points ainsi que P_n^g . Cette aire sera soustraite de l'aire déjà calculée.

Ce processus sera répété jusqu'à ce que le dernier point soit traité.

Le même processus est appliqué à la frontière droite.

La figure B1 illustre l'application de cette méthode sur un exemple d'une silhouette ayant une seule partie selon XLWDOS. Nous illustrons en jaune la surface soustraite.

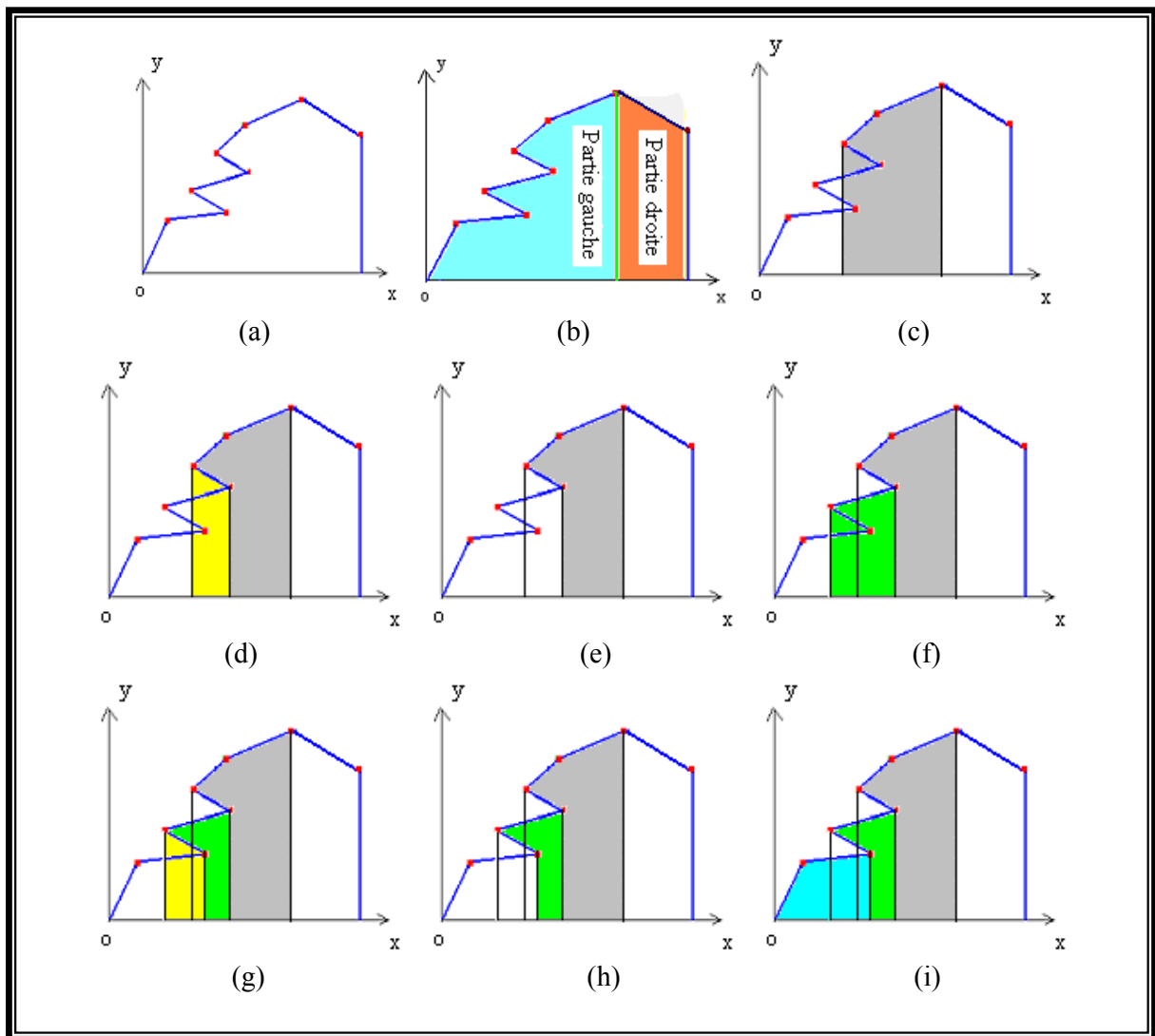


Figure B.1 Calcul de la surface d'une partie XLWDOS

B.2 Exemple de calcul de surfaces

Nous présentons ci-dessous un exemple des étapes de calcul des surfaces des parties d'une requête et des modèles concernés par la mesure de similarité (figure B.2)

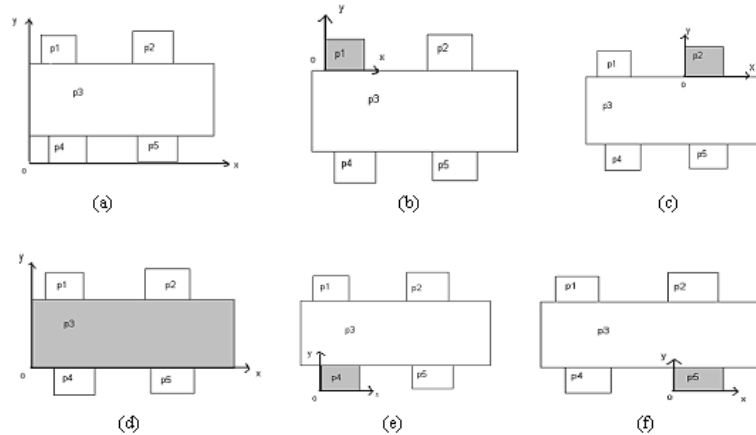


Figure B.2 Etape de calcul de la surface d'une silhouette

B.3 Exemple de classement de modèles

Nous montrons à titre d'exemple comment les modèles seront classés en fonction des mesures de similarité calculées. Le classement se fera par ordre décroissant de la ressemblance avec la requête (voir figure B.3).

Commençons par calculer le vecteur surface de la silhouette requête (figure B.3.a) :

Surface de P1=1400, Surface de P2=3185, Surface de P3=27720, Surface de P4=2107, Surface de P5=2993. Ces surfaces forment les composantes du vecteur surface de la requête Surface de (a)= (1400, 3185, 27720, 2107, 2993).

De la même manière nous calculons les vecteurs surfaces de chaque silhouette modèle de la figure B.3.

Surface de (b)= (3657, 2600, 27610, 1353, 1320)

Surface de (c)= (828, 990, 3364, 756, 825)

Surface de (d)= (1760, 910, 18966, 2665, 1540)

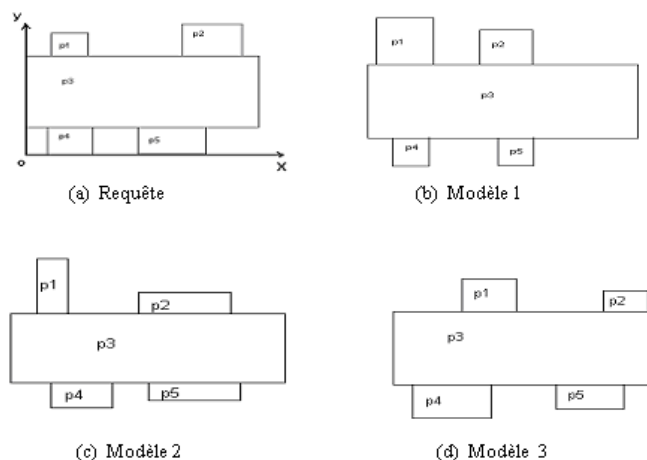


Figure B.3 Classement des modèles en fonction de la mesure de similarité

Le calcul de la mesure de similarité s'effectuera pour la requête avec chacun des trois modèles séparément et ce en utilisant la formule définie précédemment. $SF_1 = \sum_{i=1}^{NbParts} (S_{mi} - S_{ri})^2$:

La requête (a) avec le modèle1 (b) :

$$(1400-3657)^2 + (3185-2600)^2 + (27720-27610)^2 + (2107-1353)^2 + (2993-1320)^2 = 8815819.$$

La requête (a) avec le modèle1 (c) :

$$(1400-328)^2 + (3185-990)^2 + (27720-3364)^2 + (2107-756)^2 + (2993-825)^2 = 605707370.$$

La requête (a) avec le modèle1 (d) :

$$(1400-828)^2 + (3185-990)^2 + (27720-3364)^2 + (2107-756)^2 + (2993-825)^2 = 604885370.$$

D'après les résultats obtenus nous constatons que c'est la silhouette (b) qui ressemble le plus à (a) car elle minimise la différence des surfaces, donc elle en sera classée en premier, puis c'est la silhouette (d) qui prend la deuxième place, en fin la silhouette (c).

Remarque :

La surface est calculée en pixel au carré, sachant qu'un pixel=0,135 mm.

Bibliographie

- [1] Abdeslam A. Modélisation du système visuelle humain, Processus de description et de reconnaissance d'objets. Rapport de DEA Toulouse, 1985.
- [2] Aggarawal J.K. Multisensor Fusion for Computer Vision. Springer Verlag, 1993.
- [3] Alvarado C, Oltmans M, Davis R. A Framework for Multi-Domain Sketch Recognition. In the proc. Of AAAI Spring Symposium on Sketch Understanding, 2002
- [4] Aouat S. Noisy Silhouettes correction based on LWDOS. Int. wokshop on Text, image and speech recognition. Annaba 12-13 December 2005.
- [5] Aouat S, Larabi S. Comparison of silhouettes descriptions. ICGST Int. Conf. On Graphics, Vision and Image Processing. Cairo, Egypt. 19-21 December 2005.
- [6] Aouat S, Larabi S. Transformation de descripteurs de silhouettes pour la comparaison d'images. 5^{ème} Séminaire National en Informatique. 2-4 Mai 2006.
- [7] Aouat S, Larabi S. Comparison of silhouettes descriptions. Int. Journal on Graphics, Vision and Image Processing. A special Issue on "Medical Image Processing", 61-66, 2006.
- [8] Aouat S, Larabi S. Coarse Comparison of outlines shapes descriptors. Computational Vision and Medical Image Processing VipIMAGE conference, Porto, Portugal, 17-19 October 2007. (Edited by Taylor & Francis Group, London, pp 261-267, UK © 2008).
- [9] Aouat S, Larabi S. Correction of Detailed descriptions to compare noisy silhouettes. 2^{ème} Conference sur la Vision Artificielle. Université Mouloud Mammeri. Tizi Ouzou, 18-20 Novembre 2007.
- [10] Aouat S, Larabi S. La reconnaissance des silhouettes d'objets en utilisant une description textuelle. IEEE Canadian Conference on Electrical and Computer Engineering. Niagara Falls, Ontario, Canada, 4-7 May 2008.
- [11] Aouat S, Laiche N, Souami F, Larabi S. 3D object indexing and recognition. Applied Mathematics and Computaion. (© 2007 Elsevier Inc.) 196: 318-332, 2008.
- [12] Aouat S, Larabi S. Comparison of Detailed Descriptors of Noisy Silhouettes. Machine, Graphics & Vision. To appear on volume 18 Number 1, 2009.
- [13] Aouat S, Larabi S. Coarse Comparison of Silhouettes using XLWDOS Language. Int. journal for Computational Vision and Biomechanics. To appear on Volume 2 Number 1, 2009.

- [14] Arman F, Aggarawal J.K. CAD-base Vision: Object recognition in cluttered range images using recognition strategies. *Computer Vision, Graphics and Image Processing*, 58(1): 33-48, July 1993.
- [15] Asada H, Brady M. The curvature primal sketch. MIT AI Memo 1984.
- [16] ASTRÖM K. Affine and projective normalization of planar curves and regions. *Proceedings of the 3rd European Conference on Computer Vision*, Stockholm, Sweden. Mai 1994.
- [17] Ayache N. Vision stéréoscopique et perception multidimensionnelle application à la robotique mobile, collection science informatique, interne édition 1989.
- [18] Badawy O.E, Kamel M. Shape representation using concavity graphs. *Proceeding of the 16th International Conference on Pattern Recognition*, 2002.
- [19] Barrow H, Tenenbum J. *Computational Vision*. Proc. of IEEE, 69(5): 572-595, May 1981.
- [20] Belkasim S.O, Shridhar M, Ahmadi M. Pattern Recognition with Moment Invariants: a Comparative Study and New Results. *Pattern Recognition*, 24(12):1117-1138, 1991.
- [21] Belongie S, Malik J, Puzicha J. Shape Matching and Object Recognition Using Shape Contexts. *IEEE Trans.PAMI*, 24, April 2002.
- [22] Benamoun M, Boashash B. A probabilistic approach for automatic parameter selection for the hybrid edge detector. *IEICE Trans. fundamentals*, E80-A(8): 1423-1429, August 1997.
- [23] Benamoun M, Mamic G.J. *Object Recognition Fundamentals and case studies*. Springer Verlag, London, 2002.
- [24] Berchtold S, Keim D, Kriegel H.P. THE X-Tree : An index structure for high dimensional data. In *Proc. of the Inter. Conf. on Very Large Data Bases*, Bombay, India, 1996.
- [25] Berard F. *Vision par ordinateur pour l'interaction homme-machine fortement couplée*. Thèse de Doctorat de L'université Joseph Fourier, Grenoble , 1999.
- [26] Berg A.C, Berg T.L, Malik J. Shape Matching and Object Recognition using low Distorsion Correspondences. In *CVPR 2005*.
- [27] Bergevin R, Levine M. Generic object recognition: Building and matching coarse descriptions from line drawings. *IEEE Trans. PAMI*, 15(1): 19-36, January 1993.
- [28] Bernard S, Fishler M. *Stereo Vision*. In S. Shapiro editor, 1083-1090. John Wiley and Son, 1987.
- [29] Bernier T, Landry J. A new method for representing and matching shapes of natural objects. *Pattern Recognition*, 36: 1711-1723, 2003.
- [30] Besl P.J. *Machine Vision for three-dimensional scenes*. Chapter the free-form surface matching problem, Academic Press, 25-71, 1990.
- [31] Besl P.J. *Object recognition in computer vision*. Springer Verlag, 191-206, 1995.

-
- [32] Besl P.J, Jain R. Three dimensional object recognition. *Computing surveys*, 17(1): 75-145, March 1989.
- [33] Bhanu B, Faugeras O.D. Shape matching of two dimensional objects. *IEEE Trans. PAMI*, 6(2): 137-156, March 1984.
- [34] Blum H. A Transformation for Extracting New Descriptors of Shape. In Whaten-Dunn edition, pp.362-380, MIT Press, Cambridge, MA,1967.
- [35] Borotschnig H, Paletta L, Prantl M, Pinz A. Appearance based active object recognition. *Image and Vision Computing*, 18: 715-727, 2000.
- [36] Bouagar S. Utilisation de la connaissance en appariement stéréoscopique. Thèse de magister USTHB, 2003.
- [37] Bouagar S . A new approach of outline shape matching. *ICGST Int. Conf. On Graphics, Vision and Image Processing*. Cairo, Egypt. 19-21 December 2005.
- [38] Brady J, Nandhakumar N, Aggarawal J.K. Recent progress in object recognition from range data. *Image and Vision Computing*, 7(4): 295-307, November 1989.
- [39] Brooks R. Symbolic reasoning among 3D models and 2D images. *Artificial Intelligence*, 17: 285-348, 1981.
- [40] Bruce V, Green P.R, Georgeson M.A. *Virtual perception*. Psychology press, 1996.
- [41] Bukhardt B, Siggelkow S. Invariant features in pattern recognition –fundamentals and applications. In C. Kotropoulos and I. Pitas, editors, *Nonlinear Model-Based Image/Video processing and Analysis*, 269-307. John Wiley&Sons, 2001.
- [42] Carrez C. *Structure de données en Java, C++ et Ada 95, Pratique et outils de contrôle*. CNAM, Paris InterEditions, 1997.
- [43] Castan S. *Vision par ordinateur, technique de l'ingénieur*, 1985.
- [44] Chakrabarti K, Binderberger M. O, Porkaew K, Mehrotra S. Similar Shape Retrieval in MARS. In *Proc. Of IEEE Int. Conf. On Multimedia and Expo*, New York, USA, 2000.
- [45] Chakrabarti K, Mehrotra S. The Hybrid tree : An index structure for high dimensional feature spaces. In *Proc. Int. Conf. Data Engineering*, Australia, 1999.
- [46] Chellappa R, Bagdazian R. Fourier Coding of Image Boundaries. *IEEE Trans. PAMI*-6(1):102-105, 1984.
- [47] Chellappa R, Zheng Q, Burlina P, Shekhar C, Eom K. On the positioning of multi sensor image data for image exploitation and target recognition. *Proc. of the IEEE*, 65: 120-138, January 1997.
- [48] Chen Y, Medioni G. Description of complex objects from multiple range images using an infalting balloon model. *Computer Vision and Image Understanding*, 61(3): 325-334, 1995.
- [49] Chen J.L, Stockman G.C. 3D free form object recognition using indexing by contour features. *Computer Vision and Image Understanding*, 71(3): 334-355, September 1998.

- [50] Chetverikov D. A Simple and Efficient Algorithm for Detection of High Curvature Points in Planar Curves. 10th Int. Conf., Groningen, the Netherlands, August 2003.
- [51] Chien C, Agarrawal J.K. Model construction and shape recognition from occluding contours. IEEE Trans. PAMI, 11(4): 372-389, April 1989.
- [52] Chin R, Dyer C. Model_based recognition in robot vision. Computing surveys, 18: 67-108, 1986.
- [53] Chua C, Jarvis R. Point signature: A new representation for 3D object recognition. Int. journal of computer vision, 25(1): 63-85, 1997.
- [54] Cohen F, Wang J. Part I modeling image curves using invariant 3D object curve models. IEEE Trans. PAMI, 16(1): 1-12, January 1994.
- [55] Daoudi M, Matusiak S. Visual Image Retrieval by Multiscale Description of User Sketches. Journal of Visual Language and Computing, 11:287-301, 2000.
- [56] Das S, Bhanu B, Ho C. Generic object recognition using multiple representations. Image and Vision Computing, 14: 323_338, 1996.
- [57] Das M, Paulik M.J, Loh N. K. A Bivariate Autoregressive Modeling Technique for Analysis and Classification of Planar Shapes. IEEE Trans. PAMI-12(1):97-103, 1990.
- [58] Davies E. R. Machine Vision: Theory, Algorithms, Practicalities. Academic Press, 1997.
- [59] Dorai C: Cosmos. A framework for representation and recognition of 3D free form objects. PhD thesis, Michigan University, 1996.
- [60] Dorai C, Jain A.K. COSMOS a representation scheme for 3D free form objects. IEEE Trans. PAMI, 19(10): 1115-1130, October 1997.
- [61] Dubois S. R, Glanz F. H. An Autoregressive Model Approach to Two-dimensional Shape Classification. IEEE Trans. PAMI-8:627-637, 1986.
- [62] Dudani A. S, Breeding K. J, McGhee R. B. Aircraft Identification by Moment Invariants. IEEE transaction on Computers, C-26(1):39-46, 1977.
- [63] Eberly D. H. Geometric Methods for Analysis of Ridges in N-dimensional Images. PhD thesis, University of North Carolina at Chapel Hill, 1994.
- [64] Edwards J.L, Muarse H. Coarse to fine adaptive masks for appearance matching of occluded scenes. Machine vision and application, 10: 232-242, 1998.
- [65] Eom K, Park J. Recognition of Shape by Statistical Modeling of Centroidal Profile. Proc. 10 th Intl. Conf. On Pattern Recognition, Vol.1:860-864, Atlantic City, 1990.
- [66] Fan T.J. Describing and recognising 3D objects using surface properties. Springer Verlag, 1990.
- [67] Faugeras O. Three dimensional computer vision.: A geometric Viewpoint. MIT press, 1993.

-
- [68] Faugeras O, Hebert M. Techniques for 3D machine perception, Elsevier science publishers, 13-51, 1986.
- [69] Faugeras O, Toscani G. Camera calibration for 3D computer vision. Int. Workshop on Machine Intelligence, Tokyo, Japon 1987.
- [70] Finkelstein A, Salesin D. Multiresolution curves. In SIGGRAPH, Orlando, Florida, 261-267, July, 1994.
- [71] Flynn P.J, Jain A.K. Handbook of pattern recognition and image processing. computer vision, chapter three dimensional object recognition, 498-540, Academic Press, 1994.
- [72] Forsyth D, Mundy J, Zisserman A, Coello C, Heller A, Rothwell C. Invariant descriptors for 3D object recognition and pose. IEEE Trans. PAMI, 13(10): 971-991, October 1991.
- [73] Freeman H. On the Encoding of Arbitrary Geometric Configurations. IEEE Trans. on Electronic Computers, Vol. EC-10, pp. 260-268, 1961.
- [74] Fu K. S. Syntactic Methods in Pattern Recognition. Academic Press, 1974.
- [75] Gabor D. Theory of communication. Journal of the IEE, 93: 211-226, 1946.
- [76] Geiger D, Liu T, Kohn R.V. Representation and Self-Similarity of shapes. IEEE Trans. PAMI, 25(1), January 2003.
- [77] Gonzalez R.C, Woods R. E: Digital Image Processing. Ed. Addison-Wesley, 1992.
- [78] Gordon I.E. Theories of Visual Perception. John Wiley & Sons Ltd., 1989.
- [79] Goshtasby A. Description and Discrimination of Planar Shapes Using Shape Matrices. IEEE Trans. PAMI-7:738-743, 1985.
- [80] Graham R. I. An efficient algorithm for determining the convex hull of finite polar set. Information processing letters, 1972.
- [81] Greenspan M, Boulanger P. Efficient and reliable template set matching for 3D object recognition. In proc. of Int. ICDIM conf., Ottawa, Canada, 230-237, October 1999.
- [82] Grimson W.E.L. On the recognition of parameterised 2D objects. Int. Journal of computer vision, 3: 353-372, 1988.
- [83] Grimson W.E.L. On the recognition of curved objects. IEEE Trans. PAMI, 11(6): 632-643, June 1989.
- [84] Grimson W.E.L, Huttenlocher D.P. On the sensitivity of the Hough transform for object recognition. IEEE Trans. PAMI, 12(3): 255-274, March 1990.
- [85] Grimson W.E.L, Lozano-perez T. Localizing overlapping parts by searching the interpretation tree. IEEE Trans. PAMI, 9(4): 469-482, July 1987.
- [86] Gros P, Bournez O, Boyer E. Utilisation des quasi-invariants géométriques pour l'appariement et la modélisation des images de segments de droites. Rapport de recherche n°2608. INRIA. 1995.

- [87] Groskey W.I, Mehrotra R. Index-based Object Recognition in Pictorial Data Management. *Computer Vision, Graphics, and Image Processing*, 52:416-436, 1990.
- [88] Groskey W. I, Neo P, Mehrotra R. A Pictorial index mechanism for Model-based matching. *Data & Knowledge Engineering*, 8:309-327, 1992.
- [89] Gueziec A, Ayache N. Smoothing and matching of 3D space curves. *Int. journal of computer vision*, 12(1): 79-104, 1994.
- [90] Gupta A, Bajcsy R. Surface and volumetric segmentation of range images using biquadrics and superquadrics. In *proc of the Int. ICPR conf.*, Hague, Netherlands, 1992.
- [91] Haralick R, Shapiro L. *Computer and Robot Vision. Volume I and II*, Addison-Wesley, 1993.
- [92] He Y, Kundu A. 2-D Shape Classification Using Hidden Markov Model. *IEEE Trans. PAMI-13(11)*:1172-1184, 1991.
- [93] Hebert M, Ikeuchi K, Delingette H. A spherical representation for recognition of free form surfaces. *IEEE Trans. PAMI*, 17(7): 681-689, July 1995.
- [94] Hirata K, Kato T. Query by Visual Example—Content based Image Retrieval. In *Proc. of the 3rd Int. Conf. on Extending Database Technology* pp.56-71, Vienna, Austria, 1992.
- [95] Horaud R, Manga O. *Vision par ordinateur outils fondamentaux*, 2eme édition revue et augmentée, Hermes paris, 1995.
- [96] Horn B. Extended Gaussian image. *Proc. of the IEEE*, 72: 1671-1686, 1984.
- [97] Hu M. K. Visual pattern Recognition by Moment Invariants. *IRE Transactions on Information Theory*, IT-8:179-187, 1962.
- [98] Huttenlocher D.P, Klanderman G, Rucklidge W. Comparing images using the hausdorff distance. *IEEE Trans. PAMI*, 15(9): 850-863, September 1993.
- [99] Huttenlocher D.P, Lillien R, Olson C. View based recognition using an eigenspace approximation to the hausdorff measure. *IEEE Trans. PAMI*, 21(9): 951-955, September 1999.
- [100] Huttenlocher D.P, Ullman S. Recognising solid objects by alignment with an image. *Int. journal of computer vision*, 5(2): 195-212, 1990.
- [101] Jahne B. *Digital Image Processing—Concepts, Algorithms and Scientific Applications*. Springer-Verlag, Berlin, Heidelberg, 1997.
- [102] Jain A.K, Hoffman R. Evidence based recognition of 3D objects. *IEEE Trans. PAMI*, 10(6): 783-802, November 1988.
- [103] Johnson A.E, Hebert M. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Trans. PAMI* 10(5), May 1999.
- [104] Kang S.B, Ikeuchi K. The complex EGI: A new representation for 3D pose determination. *IEEE Trans. PAMI*, 15(7): 707-721, July 1993.

-
- [105] Kashyap K. L, Chellappa R. Stochastic Models for Closed Boundary analysis: Representation and Reconstruction. *IEEE Transaction on Information Theory*, 27:627-637, 1981.
- [106] Kass M, Witkin A., Terzopoulos D. Snakes: Active contour models. *Int. Journal of computer vision*, 1(4): 321-331, 1988.
- [107] Keren D, Cooper D, Subrahmonia J. Describing complicated objects by implicit polynomials. *IEEE Trans. PAMI*, 16(1): 38-53, January 1994.
- [108] Keyzers D, Deselaers T, Breuel T.M. Optimal geometric matching for patch-based object detection. *Electronic Letters on computer vision and image analysis*, 6(1): 44-54, 2007.
- [109] Kim Y.-S, Kim W.Y. Content-based Trademark Retrieval System Using a Visually Salient Feature. *Image and Vision Computing*, 16:931-939, 1998.
- [110] Kim W.Y, Kim Y.S. A Region-based Shape Descriptor Using Zernike Moments. *Signal Processing: Image Communication*, 16:95-102, 2000.
- [111] Kim D.H, Yun I.D, Lee S.U. A new shape decomposition scheme for graph-based representation. *Pattern Recognition*, 38, 2005.
- [112] Koenderink J, Van Doorn A. Surface shape and curvature scales. *Image and Vision Computing*, 10(8): 557-565, October 1992.
- [113] Lamdan Y, Wolfson H. Geometric hashing: A general and efficient model based recognition sheme. *IEEE ICCV conf.* 238-249, Florida, December 1988.
- [114] Lamiroy B. Reconnaissance et modélisation d'objet 3D à l'aide d'invariants projectifs et affines. Thèse de doctorat. Institut National Polytechnique de Grenoble, GRAVIR-IMAG-INRIA Rhône-Alpes. Juillet 1998.
- [115] Lanser S, Zierl C, Munkelt O, Radig B. MORAL: a vision based object recognition system for autonomous mobile systems. In 7th Int. ICCAIP Conf. 33-41, Kiel, September 1997.
- [116] Larabi S. Utilisation de la géométrie projective en vision par stéréoscopique binoculaire. Thèse de doctorat. Institut National Polytechnique de Toulouse. France. Avril 1991.
- [117] Larabi S. Textual description of images, *International Journal of Simulation Modelling*, 6(2), 93-102, June 2007.
- [118] Larabi S, Aouat S. Matching XML descriptors of silhouettes. *ICGST Int. Conf. On Graphics, Vision and Image Processing*. 19-21 December 2005.
- [119] Larabi S, Aouat S. Matching XML descriptors of silhouettes. *Int. Journal on Graphics, Vision and Image Processing*. A special Issue on "Applicable Image Processing", 63-68, 2006.
- [120] Larabi S, Bouagar S, Trespaderne F.M, Lopez E.F. LWDOS LAnguage for Writing Descriptors of Outline Shapes. In the 13th Scandinavian Conf. on Image Analysis, Goteburg, sweden, june 2003.

- [121] Larabi S, Bouagar S. An XML language for Writing Descriptors of Silhouettes. ICGST int. Conf. on Graphics, Vision and Image Processing Cairo 2005.
- [122] Latecki L.J, Lakamper R. Application of Planar Shape Comparison to object Retrieval in Image Databases. Pattern Recognition 35 : 15-29, 2005.
- [123] Latecki L.J, Lakamper R, Sobel M. Polygonal approximation of point sets. IWCIA 2006. In LNCS Springer Berlin/Heidelberg, book: Combinatorial images analysis, 159-173, June 2006.
- [124] Latecki L.J, Lakaemper R, Wolter D. Optimal partial shape similarity. Image and vision computing, 23(2): 227-236, 2005.
- [125] Leibe B, Schiele B. Analyzing Appearance and Contour Based Methods for Object Categorization. Int. Conf. (CVPR'03), Madison, Wisconsin, June 2003.
- [126] Leonardis A, Bishof H. Robust recognition using eigenimages. CVIU,78: 9-118, 2000.
- [127] Levine M.D. Vision in man and machine. McGraw-Hill series, New York 1985.
- [128] Liao C, Medioni G. Representation of image data with B-spline surface patches. In Int. ICPR conf. 745-748, Hague, 1992.
- [129] Liao S.X, Pawlak M. On Image Analysis by Moments. IEEE Trans. PAMI-18(3):254-266,1996.
- [130] Loncaric S. A Survey of Shape Analysis Techniques. Pattern Recognition, 31(8):983-1001, 1998.
- [131] Lowe D. Fitting parameterised 3D models to images. IEEE Trans. PAMI, 13(5): 441-450, May 1991.
- [132] Lowe D.G. Perceptual organization and visual recognition. Kluwer academic publishers, Boston 1995.
- [133] Lowe D.G. Distinctive image features from scale-invariant keypoints. Int. journal of computer vision. 60(2): 91-110, 2004.
- [134] Lu G.J, Sajjanha A. Region-based Shape Representation and Similarity measure suitable for content-based image retrieval. MULTIMEDIA Systems, 7(2):165-174,1999.
- [135] Malaviya A, Peters L. Fuzzy handwriting description language FOHDEL. Patern recognition , 33(1), 119-131, January 2000.
- [136] Mamic G, Bennamoun M. Review of 3D object representation techniques for automatic object recognition (invited paper). In int. conf. CVCIP, Perth, June 2000.
- [137] Mamic G, Bennamou M. Automatic flaw detection in textiles using a Neyman Pearson detector. In 15th Int. ICPR conf. Barcelona, September 2000.
- [138] Mamic G, Bennamoun M. Representation and recognition of 3D free form objects. Digital signal processing, 2001.
- [139] Marr D. vision. Freeman and company. San Fransisco .USA 1982.

-
- [140] Marr D, Nishihara H.K. Visual information processing: artificial intelligence and the sensorium of sight. *Technology review*, 81(1°: 26-23, October 1978.
- [141] Marthon P. Contribution à la modélisation et à la résolution du problème de la vision passive tridimensionnelle par ordinateur, *Analyse d'un monde d'objets polyédrique*.
Thèse de doctorat de l'INPT, Toulouse 1987.
- [142] Medioni G, Franois A.R.J. 3-D structures for generic object recognition. Invited conference paper, In the proc. of the Int. Conference on Pattern Recognition, 1: 30-37, Barcelona, Spain, September, 2000.
- [143] Mehtre B.M, Kankanhalli Lee, W. F. Shape Measures for Content based ImageRetrieval: A Comparison. *Information Processing & Management*, 33(3):319-337, 1997.
- [144] Mikolajczyk K, Shmid C. A performance evaluation of local descriptors. *IEEE Trans. PAMI*, 27(10), 1615-1630, 2005.
- [145] Ming J, Bhanu B. ORACLE: An integrated learning approach for object recognition. *Int. journal IJPRMI*, 11(6): 961-990, 1997.
- [146] Mohamad D, Sulong G, Ipson S.S. Trademark Matching Using Invariant Moments. *Proc. of the Second Asian Conf. on Computer Vision*, pp.439-444, Singapore, December 5-8, 1995.
- [147] Mokhtarian F. Silhouette based occluded object recognition through curvature scale space. *Machine Vision and Application*, 10: 87-97, 1997.
- [148] Mokhtarian F, Abasi S. Shape similarity retrieval under affine transforms. *Pattern Recognition* 35(1): 31-41, January 2002.
- [149] Mokhtarian F, Mackworth K. A Theory of Multiscale, Curvature-based Shape Representation For Planar Curves. *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 12 (8), 1992.
- [150] Morse B.S. Computation of Object Cores From Grey-level Images. Ph.D thesis, University of North Carolina at Chapel Hill, 1994.
- [151] Muarse H, Nayar S. Visual learning and recognition of 3D objects from appearance. *Int. journal of computer vision*, 14: 5-24, 1995.
- [152] Mundy J.L, Zisserman A. *Geometric invariance in Computer Vision*. MIT Press, 1992.
- [153] Nalwa V. *A guided tour of computer vision*. Addison Wesley Publishing, 1993.
- [154] Nevatia R, Binford T. Description and recognition of curved objects. *Artificial intelligence*, 8:77-98, 1977.
- [155] Niblack W, Barber R, Equitz W, Flickner M, Glasman E, Petkovic D, Yanker P, Faloutsos C, Taubin G. The QBIC Project: Querying Image By Content Using Color, Texture and Shape. In *Proc. SPIE Storage and Retrieval for Image and Video Databases*, vol.1908, pp.173-187, 1993.

- [156] Nowak E, Jurie F, Triggs B. Sampling strategies for bag-of-features image classification. ECCV, Springer, 2006.
- [157] Ohba K, Ikeuchi K. Detectability, uniqueness and reliability of eigen windows for stable verification of partially occluded objects. IEEE Trans. PAMI, 19(9): 1043-1048, September 1997.
- [158] Orrite C, Herreo J.E. Shape matching of partially occluded curves invariant under projective transformation. Computer Vision and Image Understanding, 93(1), 2004.
- [159] Otterloo P. J. A contour-Oriented Approach to Shape Analysis. Prentice Hall Int. (UK) Ltd. pp.90-108, 1991.
- [160] Pavlidis T. Algorithms for Graphics and Image Processing. Computer Science Press, pp.143,1982.
- [161] Pentland A. Perceptual organisation and the representation of natural form. Artificial intelligence, 28: 293-331, May 1986.
- [162] Pentland A, Sclaroff S. Closed form solutions for physically based shape modeling and recognition. Technical report, MIT Media lab Vision and modelling group, March 1990.
- [163] Petrakis E.G.M, Diplaros A, Milios E. Matching and Retrieval of Distorted and Occluded Shapes Using Dynamic Programming, IEEE Trans. on PAMI, 24(11) November 2002.
- [164] Philbin J, Chum O, Isard M, Sivic J, Zisserman A. Object retrieval with large vocabularies and fast spatial matching. Computer Vision and Pattern Recognition, 1-8, June 2007.
- [165] Ponce J.P, Brady M. Three dimensional Machine Vision , chapter toward a surface primal sketch, 195-240. Kluwer series in engineering and science, 1987.
- [166] Ponce J.P, Keriven R , Faugeras O.D. Multi-view stereo reconstruction and scene flow estimation with a global image -based matching score. Int. Journal of computer vision, 72(2): 179-193, 2007.
- [167] Pope A.R. Model_based object recognition; A survey of recent research. Technical report, University of British Columbia, January 1994.
- [168] Prokop R. J, Reeves A. P. A Survey of Moment-based Techniques for Unoccluded Object Representation and Recognition. Graphical Models and Image Processing, Vol.54, pp.438-460,1992.
- [169] Ramanan D. Tracking people and recognizing their activities. Phd dissertation, Univ. of California, Berkeley, 2005.
- [170] Ramanan D. Tracking people by learning their Appearance. IEEE Trans. PAMI, 29(1): 65-81, 2007.
- [171] Rigoutsos I, Hummel R. A Mayesian approach to model matching with geometric hashing. Computer vision and image understanding, 62(1): 11-26, July 1995.

-
- [172] Rivlin E, Weiss I. Deformation invariants in object recognition. *Computer vision and image understanding*, 65(1): 95-108, January 1997.
- [173] Roberts L. *Optical and Electro-Optical Information Processing*, Chapter Machine Perception of three dimensional solids. MIT Press Cambridge, 1965.
- [174] Sajjanhar A. A Technique for Similarity Retrieval of Shapes. Master thesis, Monash University, Australia, 1997.
- [175] Samet H. *Design and analysis of spatial data structures*. Addison-Wesley, 1990.
- [176] Scassellati B, Slexopoulos S, Flickner M. Retrieving Images by 2D Shape: A Comparison of Computation Methods with Human Perceptual Judgments. In *SPIE Proc. Storage and Retrieval for Image and Video Databases II*, Vol.2185, pp.2-14, 1994.
- [177] Sebastian T, Klein P, Kimia B. Recognition of Shapes by Editing Their Shock Graphs. *IEEE Trans. PAMI*, 26(5): 550-571, may 2004.
- [178] Sekita I, Kurita T, Otsu N. Complex Autoregressive Model for Shape Recognition. *IEEE Trans. PAMI-14*:pp489-496, 1992.
- [179] Sethi A, Renaudie D, Kriegman D, Ponce J. Curve and surface duals and the recognition of curved 3D objects from their silhouettes. *Int. Journal of computer vision* 58 (1), 73-86, 2004.
- [180] Shao Z, Kittler J. Shape representation and recognition based on invariant unary and binary relations. *Image and Vision Computing*, 17: 429-444, 1999.
- [181] Shapiro L.G. *Techniques for 3D machine perception*, 255-270, Elsevier Science publishers, 1986.
- [182] Shapiro L.G, Haralick R.M. Organisation of relational models for scene analysis. *IEEE trans. PAMI*, 4(6): 595-602, November 1982.
- [183] Shekhar C, Govindu V, Chellapa R. Multisensor image registration by feature consensus, *Pattern recognition*, 32 : 39-52, January 1999.
- [184] Shirai Y. *Three dimensional computer vision*. Springer verlag, 1987.
- [185] Solina F. Shape recovery and segmentation with deformable parts models. PhD thesis, University of Pennsylvania, 1987.
- [186] Sonka M, Hlavac V, Boyle R. *Image processing, Analysis and machine vision*. Thomson Computer Press, 1993.
- [187] Squire D. M, Caelli T. M: *Invariance Signature. Characterizing Contours by Their Departures from Invariance*. *Computer Vision and Image Understanding*, 77:284-316, 2000.
- [188] Stein F, Medioni G. Structural indexing: Efficient 3D object recognition. *IEEE Trans. PAMI*, 14(20): 125-145, February 1992.
- [189] Stockman G. Object recognition and localisation via pose clustering. *Computer vision, graphics and image processing*, 40: 361-387, 1987.
- [190] Stollintz E, Deroose T, Salein D. *Wavelets for computer graphics: Theory and applications*. Morgan-Kaufman, 1996.

- [191] Sullivan S, Ponce J. Automatic model construction and pose estimation from photographs using triangular splines. *IEEE Trans. PAMI*, 20(10): 1091-1096, October 1998.
- [192] Sumi Y, Kawai Y, Yoshimi T, Tomita F. 3D object recognition in cluttered environments by segment based stereo vision. *Int. Journal of Computer Vision*, 46(1), 2002.
- [193] Taubin G. Estimation of planar curve, surfaces, and non planar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Trans. PAMI*, 13(11): 1115-1137, November 1991.
- [194] Taubin G, Cooper D. B. Recognition and Positioning of Rigid Objects Using Algebraic Moment Invariants. *SPIE Conf. On Geometric Methods in Computer Vision*, Vol.1570, pp.175-186, 1991.
- [195] Taubin G, Cooper D.B. Object Recognition Based on Moment (or Algebraic) Invariants. In J. Mundy and A. Zisserman ed. *Geometric Invariance in Computer Vision*, pp.375-397, 1992.
- [196] Taza A, Suen C. Discrimination of Planar Shapes Using Shape Matrices. *IEEE Transaction on Systems, Man and Cybernetics*, 19:1281-1289, 1989.
- [197] Teague M.R. Image Analysis Via the General theory of Moments. *Journal of Optical Society of America*, 70(8):920-930, 1980.
- [198] Teh C.H, Chin R. T. On Image Analysis by the Methods of Moments. *IEEE Trans. PAMI*- 10(4):496-513, 1988.
- [199] Terzopoulos D, Metaxas D. Dynamic 3D models with local and global deformations: Deformable superquadratics. *IEEE Trans. PAMI*, 13: 703-714, 1991.
- [200] Toutin T, Vester C. Radar et stéréoscopie. Centre canadien de télédétection (www.ccrs.nrcan.gc.ca). 15-12-2003.
- [201] Turk M, Pentland A. Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1), 1991.
- [202] Ullman S, Basri R. Recognition by linear combination of models. *IEEE Trans. PAMI*, 13(10): 992-1006, October 1991.
- [203] Vaillant R, Faugeras OD. Using extremal boundaries for 3D object modelling. *IEEE Trans. PAMI*, 14(2): 157-173, February 1992.
- [204] Vijayakumar B, Kriegman D, Pnce J. Structure and motion of curved 3D objects from monocular silhouettes. *CVPR*, 1996.
- [205] Wang G, Zhang Y, Fei-Fei L. Using dependent regions for object categorization in a generative framework. *IEEE proc. CVPR* , 1597-1604, Washington, DC, USA, 2006.
- [206] Weber R, Schek H.J, Blott S. A quantitative analysis and performance study for similarity-search methods in high dimensional spaces. *Proc. of the 24th VLDB Inter. Conf. on Very Large Data Bases*, New York, US, August 1998.

-
- [207] Wechsler H. Computational vision. Academic press, New York, 1990.
- [208] Weiss I, Ray M. Model-based recognition of 3D objects from single images. IEEE Trans. PAMI, 23(2): 116-128, February 2001.
- [209] Wong A, Lu S, Rioux M. Recognition and shape synthesis of 3D objects based on attributed hypergraphs. IEEE Trans. PAMI, 11(3): 279-290, 1989.
- [210] Wu X, Bhanu B. Gabor wavelet representation for 3D object recognition. IEEE Trans. PAMI, 6(1): 47-64, January 1997.
- [211] Yang X, Adluru N, Latecki L.J, Bai X, Pizlo Z. Symetry of shapes via self similarity. ISVC 2008, In LNCS springer Berlin/ Heidelberg, 5359: 561-570, in a Book Advances in visual computing. December 2008.
- [212] Yang X, Bai X, Latecki L.J, Tu Z. Improving shape retrieval by learning graph transduction. ECCV 2008, In LNCS springer Berlin/ Heidelberg, 5305: 788-801, in a Computer vision. October 2008.
- [213] Yong I, Walker J. Bowie J. An Analysis Technique for Biological Shape. Computer Graphics and Image Processing, Vol.25, pp.357-370, 1974.
- [214] Yu L, Wang R. Shape representation based on mathematical morphology. Pattern Recognition Letters, 2004.
- [215] Zabulis X, Sporring J, Orphanoudakis S.C. Perceptually relevant and piecewise linear matching of silhouettes. Pattern Recognition, 38, 2005.
- [216] Zaeri N, Mokhtarian F, Cherri A. Binarized eigenphases applied to limited memory recognition systems. Pattern analysis and application, Springer London, 11(3): 373-383, July 2008.
- [217] Zhang D. *Image Retrieval Based on Shape*. Degree of Doctor of philosophy In the Faculty of Information Technology, Monash University, PP29-55, March, 2002.
- [218] Zhang D, Lu G. Review of shape representation and description techniques. Pattern Recognition, 37(1):1-19, 2004.
- [219] Zisserman A, Forsyth D, Mundy J, Rothwell C, Liu J, Pillo N. 3D object recognition using invariance. Artificial Intelligence, 78: 239-288, 1995.